11-2015

Toward Real-Time Video-Enhanced Augmented Reality for Medical Visualization and Simulation

Alexander Maxwell Bensch
amb2189@rit.edu

Toward Real-Time Video-Enhanced Augmented Reality for Medical Visualization and Simulation

by

Alexander Maxwell Bensch

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering in the Kate Gleason College of Engineering, Department of Computer Engineering, Rochester Institute of Technology

Supervised by

Assistant Professor Dr. Cristian Linte
Department of Biomedical Engineering
November 2015

Approved by:

Dr. Cristian Linte, Assistant Professor
Thesis Advisor, Department of Biomedical Engineering

Dr. Raymond Ptucha, Assistant Professor
Committee Member, Department of Computer Engineering

Dr. Andreas Savakis, Professor
Committee Member, Department of Computer Engineering Department

Thesis Release Permission Form

Rochester Institute of Technology
Kate Gleason College of Engineering

Title:

Toward Real-Time Video-Enhanced Augmented Reality for Medical Visualization and Simulation

I, Alexander Maxwell Bensch, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

Alexander Maxwell Bensch

Date

Dedication

I dedicate this thesis to my family all those who were always there to help when the going got hard. A special thanks to my parents, John and Jannah Bensch who always pushed me to focus on my education and encouraged my endless interest in electronics. A big thank you to my friends in Rochester who never let me down; Ryan Muchard and his family for opening their home to me when I had nowhere else to turn; Rachel Sabo for being one of my closest friends here even after everyone else had graduated and gone off to greener (and warmer) pastures; and most of all Juliana Compton for being the most supportive and wonderful partner one could ask for. Thank you for believing in me even when I doubted myself.

Acknowledgments

A giant thank you to my advisor, Professor Cristian Linte, who helped me throughout the entire development of this work, coordinating flights, organizing my conference presentation, and generally making sure that I didn't get lost along the way. I couldn't have asked for a better advisor. To Kfir Ben-Zikri, a hearty thanks for breaking me out of a mental block and pointing me in the right direction. Had you not come along I would still be stuck trying to figure out what an extrinsic matrix was. To Michael Potter and Alexander Dawson-Elli, you both deserve as much credit as I do. Your initial work figuring out 3D Slicer and developing the first module was crucial to the development of this research. My deepest gratitude goes to all those who sacrificed their time to assist me in completing this work. I couldn't have done it without you.

Glossary

$^{S}\mathbf{T}_{\mathbf{D}}$ A 4x4 homogeneous matrix transforming three-dimensional space S to three-dimensional space D. v

AR Tag A specialized tag used in certain camera-based tracking applications to track individual objects.. 13

ArUco A marker tracking library designed to allow implementation of augmented reality systems using only a webcam.. 57

Extrinsic Matrix Test. 34, 58

Focal Plane Test. 33

Image Plane A two-dimensional coordinate system containing all image points $\mathbf{x}$. 20, 32

Intrinsic Matrix Test. 33

Pivot Calibration A registration process that localizes the tip of a pointer tool with reference to an attached marker.. 18, 40

Acronyms

2D Two Dimensional. 19–21, 31, 60, 61, 63, 75

3D Three Dimensional. 8, 14, 18, 20–22, 31, 60, 61, 63, 66, 68, 75

AR Augmented Reality. 3–9, 12, 18, 20, 23, 24, 26, 55–57, 72–75, 79

CAD Computer Aided Design. 55, 63

CCD Charged-Coupled Device. 15

CCS Camera Coordinate System. 32–35, 57, 61

CT Computed Tomography. 7, 14, 24, 25, 28–30, 43, 57

FLE Fiducial Localization Error. 27, 44, 45, 65

FRE Fiducial Registration Error. 27, 41, 44, 45, 65

GPS Global Positioning Satellite. 3

HMD Head-mounted Display. 5, 8–12, 74

ICP Iterative Closest Point. 19

IGT Image-Guided Therapy. 29, 40, 42, 43

IR Infrared. 15, 16, 57, 61, 71

ITK Insight Segmentation and Registration Toolkit. 27

LED Light-Emitting Diode. 15

MR Magnetic Resonance. 25, 79

MRI Magnetic Resonance Imaging. 6, 14

PCE Pivot Calibration Error. 44, 45, 65

PnP Perspective-n-Point. 20, 22, 61, 63, 74

RMS Root Mean Square. xiii, 40, 44, 46, 48, 63, 66, 68, 71

TRE Target Registration Error. 27, 44–46, 48–50, 65, 68, 70

VR Virtual Reality. 9

VRE Video Registration Error. xiii, 48, 49, 66, 68, 69, 71, 73, 74

VTK Visualization Toolkit. 27–30, 38, 57, 62

WCS World Coordinate System. 32, 34, 35, 57

Abstract

Toward Real-Time Video-Enhanced Augmented Reality for Medical Visualization and Simulation

Alexander Maxwell Bensch

Supervising Professor: Dr. Cristian Linte

In this work we demonstrate two separate forms of augmented reality environments for use with minimally-invasive surgical techniques. In Chapter 2 it is demonstrated how a video feed from a webcam, which could mimic a laparoscopic or endoscopic camera used during an interventional procedure, can be used to identify the pose of the camera with respect to the viewed scene and augment the video feed with computer-generated information, such as rendering of internal anatomy not visible beyond the image surface, resulting in a simple augmented reality environment. Chapter 3 details our implementation of a similar system to the one previously mentioned, albeit with an external tracking system.

Additionally, we discuss the challenges and considerations for expanding this system to support an external tracking system, specifically the Polaris Spectra optical tracker. Because of the relocation of the tracking origin to a point other than the camera center, there is an additional registration step necessary to establish the position of all components within the scene. This modification is expected to increase accuracy and robustness of the system.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Research Motivations

1.1.1 Minimally Invasive Surgery

Over the course of the decade between 1999 and 2009 the number of deaths per 100,000 persons in the United States due to surgical complications decreased by 39% for patients aged ≥ 85 years; 37% for patients aged 75-84 years; 38% for patients aged 65-74 years; and 28% for patients aged 45-64 years [1]. Given that minimally invasive techniques have been shown to reduce patient recovery time, blood loss, and tissue damage [2, 3], some portion of this reduction can likely be attributed to the increase in widespread adoption of minimally invasive techniques over the period 1999-2009, with the proportion of methods such as distal pancreatectomies using minimally invasive approaches tripling from 2.4% to 7.3% across the same period [4].

As a concrete example, minimally invasive cardiac therapeutic interventions are coming to the forefront to replace the typically invasive procedures where the chest is opened and the patient is placed on cardiopulmonary bypass. Recently, through the use of a Universal Cardiac Introducer[5], it has been demonstrated the ability to introduce and manipulate instruments

through the heart wall, and to effect interventions such as cryo- or RF ablations [6], or mitral-valve replacement. However, through these initial studies, the severe limitations of 2D ultrasound as a navigation tool have also been demonstrated. The obvious problem is that the 2D ultrasound image, acquired from a trans-esophageal transducer placed behind the heart, is inadequate to visualize the target and instruments with sufficient clarity to ensure the therapy is performed on target.

It is particularly difficult to gauge the 3D spatial relationship between the tools and the surrounding organs using the medical images available for visualization of the surgical scene [7], especially when 2D images such as ultrasound or X-ray are utilized. In addition, the discomfort and unfamiliarity surgeons have with these systems due to this spatial disconnect is reflected in a significant under-utilization of minimally invasive techniques across the United States [8].

As postured by *Cooper et. al*, such a widespread latency to adopt these techniques is likely due to a "lack of exposure" of surgeons during training in their residencies. As it has been shown that surgeons require an acclimation period to laparoscopic techniques [9], they argue for more focus on minimal invasion in medical training programs along with a standardization of techniques to facilitate faster adoption. However, while *Cooper et. al* seek to acclimate more surgeons to the spatial disconnect created by these surgical techniques, the research presented in this paper focuses on eliminating the disconnect entirely. Their method entails the use of an augmented-reality approach to create a virtual environment that can replicate in 3D the entire surgical scene from medical data.

1.1.2 History of Augmented Reality

In minimally invasive surgical interventions direct visualization of the target area is often not available. Instead clinicians rely on images from various sources, along with surgical navigation systems, for guidance. These spatial localization and tracking systems provide information similar to that of the Global Positioning Satellite (GPS) systems that many are familiar with. In parallel with the aspirations toward less invasive therapy delivery on the medical side, many engineering applications have been employing computer-generated models and graphics to design, simulate and visualize the interaction between different components within an assembly prior to the global system implementation. One such approach has focused on complementing the users visual field with the necessary information that facilitates the performance of a particular task a technique broadly introduced and described by *Milgram et al.* as augmenting natural feedback to the operator with simulated cues [10] and later becoming known as Augmented Reality (AR).

While the term "augmented reality" may be used somewhat loosely and in a more inclusive sense than it had originally been intended, it refers to visualization environments that combine some kind of real-world visualization with some extent of virtual or computer-generated information aimed to show what the real world view cannot, resulting in a more comprehensive, enhanced view of the world. Therefore, the term mixed reality has been suggested as a better descriptor of such environments, as, depending on the extent of real and computer-generated information, the resulting environments can lie anywhere on the spectrum of the reality-virtuality continuum shown in **Fig.** 1.1.

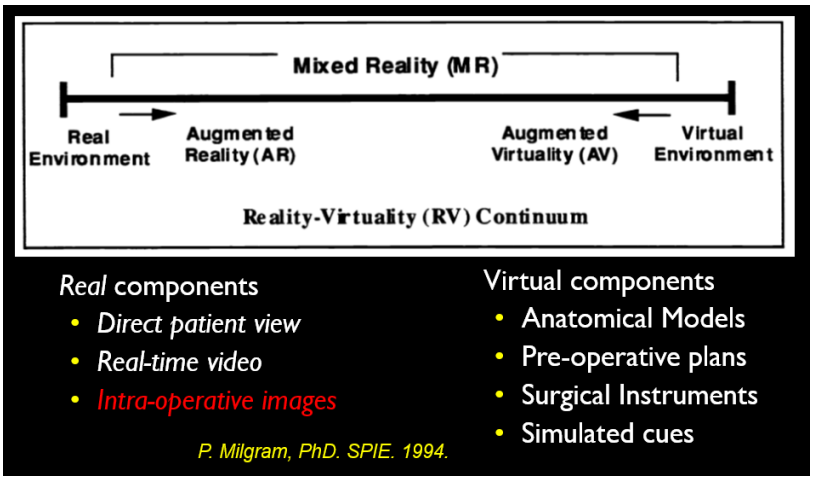

Figure 1.1: Milgram's [10] diagram depicting the components of a mixed reality environment in terms of its consistency of real and virtual elements, accompanied by real and virtual examples in image-guided interventions.

The real component of a typical mixed reality environment may consist of either a direct view of the field observed by the users eyes (i.e. optical-based AR), or a view of the field captured using a video camera and displayed to the user video-based AR. As synthetic (i.e. computer-generated) data is added to the environment, the mixed reality may become less of a traditional augmented reality and more of an augmented virtuality, yet still remain sufficiently different from a fully immersed virtual reality environment, in which the user has no access to the real-world view.

Although it is hard to determine the absolute first implementation of an AR system, one of the first industry applications of computer-based AR explicitly referred to as an "augmented reality" device was designed and implemented for industrial applications in 1994 by Tom Caudell, an engineer working in Boeing's Computer Service's Adaptive Neural Systems Research and Development Project [11]. While at first computer-generated models were displayed on computer screens and available to the workers as "guides", Caudell's revolutionary approach, based on the work of

Ivan Sutherland in 1968 [12], resorted to the use of a see-through display mounted on a head-set device work by the workers that enabled the superposition of computer models on to the real view of the physical parts hence facilitating the workers task by truly augmenting the users view with computer-simulated cues. This would be the first industrial application of Sutherland's concept of a Head-mounted Display (HMD) [12], a tool that is only just recently being explored for commercial use by companies like Oculus VR and Valve Software.

Visualization environments soon experienced some traction in the medical world, motivated primarily by the movement towards minimally invasive surgeries whose goal was to improve clinical outcome and safety by reducing procedure morbidity, recovery time and associated costs. The challenges arising alongside the trend toward minimally invasive procedures quickly revealed themselves, in terms of surgical navigation and target tissue manipulation under restricted access conditions and limited visualization, raising the need for adequate intuitive visualization critical for the performance of the procedure. The benefit of an AR system in this environment is clear, as minimizing the amount of tissue exposure has the twofold effect of minimizing visibility of the surgical site as well. Having this information present using virtual displays while still reducing tissue affectation would be greatly beneficial to the patient.

1.1.3 Applications

There are numerous applications to AR in multiple fields and industries. Ronald T. Azuma details many of these applications in his meta-survey titled "A Survey of Augmented Reality" [13]. In this survey, Azuma discussed

the benefits of the partial augmentation provided by AR as opposed to the replacement of reality provided by a fully virtual reality system. Whereas fully virtual systems have little connection to the real world (such as virtual reality-enabled video games), AR applications can provide information that enhances the user's ability to perceive and interact with the surrounding environment.

One of the most common examples of practical application is in the field of medicine. During surgery, whereas the human eye would be able to detect some features that an imaging device (e.g. a Magnetic Resonance Imaging (MRI) or ultrasound device) could not, and vice versa, an augmented reality system would allow the surgeon access to both of these types of information. Should the surgeon require a medical image of the site, a simple press of a button could immediately display such information in a contextually relevant manner. This feature also allows to give the doctor what Azuma refers to as "X-ray vision" inside the patient [13]. Using medical imaging data such as an MRI scan, which captures information from within the body, a doctor can view a patient's internal anatomy in real time. This technique could provide massive benefits to minimally-invasive surgical techniques, a goal that the work presented in this thesis attempts to address to a limited extent. Augmented Reality (AR) also has the capability of being used for medical training in a similar fashion, providing helpful instructions, identifying organs or internal elements of the body, and providing a more hands-on experience than current virtual trainers.

There have been a number of applications of this technology in recent years, each taking a different approach to scene augmentation. One of the earlier systems was the microscope-assisted guide interventions (MAGI)

system [14]. Developed in 2000, the MAGI tool provided an automated registration process for aligning a magnified microscope image onto a view of the patient. The quality of augmentation was quite high, ranging from .5mm to 1mm under specific conditions. These conditions required implanted bone markers in the patient, an obviously invasive procedure. As an alternative, the researchers also implemented a less-invasive technique that required less surgical intervention but performed with reduced accuracy, resulting in .5 to 4mm alignment accuracy.

A similar system called the Scopis Hybrid Navigation (SCOPIS GmbH, Germany) [7] provides augmentation of endoscopic video via CT volumes. The endoscope is calibrated and then tracked using either optical or magnetic methods. As endoscopic video suffers from severe barrel distortion due to lens properties necessary for a wide-angle view, there are two possible ways of handling augmentation. The endoscopic image can be undistorted using the camera's intrinsic properties, or the augmenting data can be distorted using the same information. The developers of the Scopis system opted to distort the augmenting data, as it is less computationally intensive in addition to fitting the distorted format surgeons are used to using.

One of the more unique approaches to medical augmentation is the declipseSPECT (SurgicEye GmbH) [7]. This system, while not primarily an AR system, uses augmentation to guide clinicians for data acquisition and diagnostics. What is unique about this method is that it does not use acquired fiducials. Instead, a gamma tracer is injected into the patient's bloodstream which binds to tumor cells in high concentrations. This allows to use of a gamma probe to detect the radioactive material and reconstruct it into

a 3D volume used for augmentation. After this step, the actual augmentation is performed using an external optical tracker in a similar fashion to the MAGI system.

These systems provide an example of different methods for acquiring the augmentation of the scene. However, there have also been developments in recent years regarding the display of these scenes, specifically with regards to immersion. The recent developments in commercial HMD technology, specifically the Oculus Rift (Oculus VR, USA), the HTC Vive (HTC Corporation, USA), and the Sony Morpheus (Sony, Japan) have allowed more developers to experiment with display techniques that were previously unexplored or uncommon.

One of the most widely known examples of such a technique is the da Vinci robotic operating system, a remote surgical device that allows clinicians to perform operations from across the globe using robotic-assisted procedures [15]. What is significant about da Vinci with regards to Augmented Reality is that the system uses a binocular video feed from an endoscope viewing the patient. The binocular view allows for a highly immersive view of the surgical scene, similar to the benefits provided by an HMD. There have been various attempts to introduce augmented data into this view, including overlays of coronary trees during cardiac surgery, kidney and collecting systems during partial kidney resections, and blood vessel overlays for liver tumor resection. While the techniques used for the augmentation in these cases were not necessarily unique in their own right, the popularity of da Vinci as a research platform is an indicator of the possible benefits to be reaped from binocular imaging methods in the field of AR.

With respect to contemporary work with HMDs in Augmented Reality, there are only a few examples, as such systems were not widely available until recently with the aforementioned release of commercial technologies at an affordable cost [16]. Additionally, hardware had not advanced to a level in which it was feasible to implement a low-latency, high-resolution HMD until recently. This is supported by the research of Keller et al. in 2008 [16], who states that "cost, bulk, complex infrastructure, as well as two remaining, inherent problems with optical see- through HMDs: Relative lag and registration of virtual images to real ones" were barriers for research of this kind at the time. That is not to say that there are no examples of these types of systems. The Nomad HMD was a see-through headset that provided optical overlays for a single eye. In 2013, Abe et al. The university of Central Florida developed a headset for medical purposes [16]. In 2013, [17] demonstrated virtual protractor with augmented reality (VIPAR), a needle trajectory guidance system for use in percutaneous vertebroplasty (PVP). However, as the technology powerful enough to render a high quality AR environment binocularly is still fairly new, examples of this type of application in the medical field are scarce.

Outside of the medical field, AR systems are also used for a multitude of applications by military aviators. The CAE system is used in military aircraft simulators and trainers, although its size makes it impractical for use in the field [18]. Military aircraft use alternate systems to superimpose graphics upon the pilot's view. These head-up displays provide vital information ranging from navigation and flight data to target registration and weapon targeting. One example of this is the "slaving" of a gunship's main turret to the pilot's helmet, enabling the pilot to aim the weapon by simply looking

at a target [13].

AR is also being implemented in the entertainment field. One widespread use is the "green screen effect", in which a scene is filmed in front of a large blue or green screen. This scene is then augmented during the post-processing steps to make it seem as though the recorded events occurred in another location. This technique is popular in modern films, as it is less expensive than filming on-location or creating a convincing physical set.

For a more commercial entertainment application, Augmented Reality (AR) air hockey table was developed by Mixed Reality Labs and demonstrated at the SIGGRAPH'98 conference. Featuring two HMDs, components of an air hockey game - pucks and paddles - were rendered virtually and registered to a real table [18].

1.1.4 Approaches to Augmentation of a Real Scene

Display Format

As Virtual Reality (VR) and Augmented Reality (AR) applications provide information through a visual medium, a variety of different display methods have been developed to enhance the experience had by the user of these systems. Forms of displays can be broken into two categories: Head-mounted Display (HMD)s and non-HMDs.

Head-Mounted Displays: HMDs, as previously mentioned, were first used in the 1960s [18] by Ivan Sutherland [12]. Sutherland's display was not only the first HMD, but the first optical see-through HMD. This type of display uses a semi-transparent surface to allow light from the real environment to pass through to the user's eyes. Virtual data is displayed on this surface as

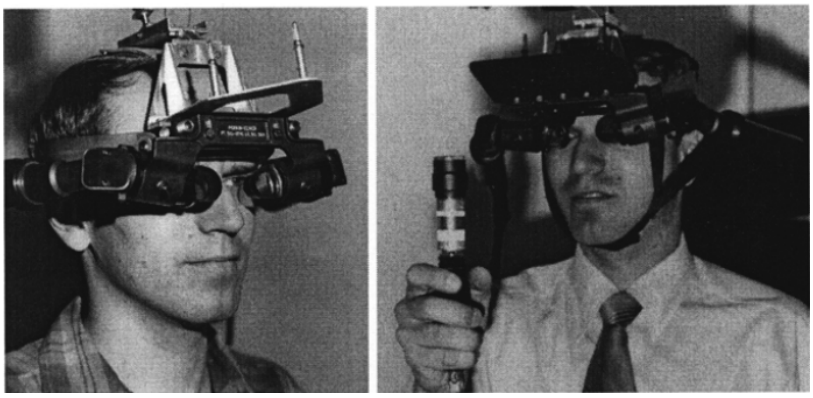

Figure 1.2: Two images of Ivan Sutherland's [12] optical see-through headmounted display. The system used two CRTs mounted on either side of the head to display information that was then reflected by a half-silvered screen in front of the eye, providing visualization of both the real and virtual data simultaneously.

well, thus allowing visualization of both the real and virtual data simultaneously. Each eye in this configuration has a dedicated display, as the view of the world varies from left eye to right eye. Sutherland's system utilized cathode-ray tube (CRT) displays mounted next to the user's eyes to emit light that was then reflected by the semi-transparent surface. Additionally, half-silvered prisms were used as the display surface for each eye to allow outside light and light from the CRTs to be viewed simultaneously. More recent implementations have improved upon the technologies used for display as well as the optical methods used to align the real and virtual data.

The most difficult aspect of using this type of HMD is obtaining an accurate registration between the real and virtual image data both spatially and temporally. The head has the capability of move very rapidly relative to the surrounding environment, thus placing a heavy constraint on processing time. Additionally, the intensity of light presents a significant obstacle to overcome, as ambient light (or a lack thereof) can result in the user having difficulty visualizing the virtual data.

The alternative to the optical see-through technique is the video see-through HMD. In this configuration, one or more (generally two) cameras are used to capture a view of the real world. This view is then augmented with virtual data and displayed on one or more screens placed in front of the user's eyes. Unlike the other method, video see-through displays require external processing to combine this data into a single format.

There are a multitude of difficulties that come with this form of display. Without a camera with optical properties customized and selected based upon the displays used, the disparity between the field of view of each camera and the field of view of the display can cause image distortion. The positioning of the cameras also is problematic, as it is physically impossible to center the camera origin at the same place where the user's view originates. Thus cameras must be offset from the eyes, causing a sense of displacement when the user views the world from this new position.

Non-Head Mounted Displays: While HMDs provide a high-level of immersion, they are currently not feasible in all scenarios. Constraints such as cost, limitations of available technologies, and spacial constraints can prevent the use of such devices in a given space. For this purpose, techniques have been developed to use common display techniques from other mediums such as television and desktop computing to compensate. Generally these methods are similar in fashion to either the optical see-through or video see-through HMDs, but are configured in such a way that they do not provide a stereo view of the world. These types of devices include half-silvered mirrors, computer monitors, and projectors. Additionally, conceptual technologies

like the UNC "Office of the Future" [18] can be argued to provide an Augmented Reality (AR) experience.

Camera-Based Tracking

An AR system is generally comprised of three key systems, those being the tool used for rendering the displayed virtual data, a device or set of devices that capture the real-world scene (e.g. a camera or camera array), and a set of devices that establish a reference frame from which the augmentation poses are determined. Among these elements, there are two primary configurations available for visually augmenting a scene with virtual data. The first of these configurations places the responsibilities of both tracking and data capture solely upon the capture system. In the case that this capture equipment is comprised of one or more cameras, the result is a camera-based tracking system.

There are various ways camera data can be used to track objects within a scene. One of these methods is the inclusion of special tags referred to as AR Tags in the tracked scene. These tags are printed with a pattern that adheres to some predetermined format and which is able to be detected by image processing tools analyzing the captured images. One implementation of this technique is the ArUco Library [19] which is utilized in this work.

The ArUco library was created with the goal of creating a technology that could efficiently track multiple markers efficiently by using a stochastic approach instead of a "complete evaluation of the search space". The library stores m markers made up of a $(n+2)*(n+2)$ pixelated, black-and-white grid (Seen in **Fig.** 1.3). An outside border two pixels wide aids detection by isolating the inner nxn grid. This inner grid is used to identify each

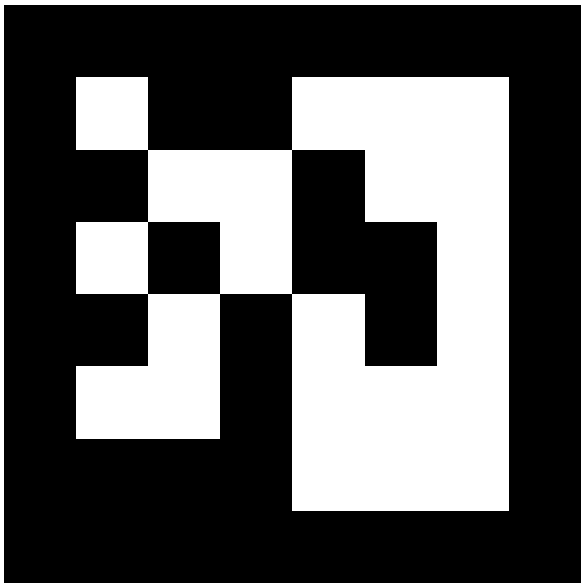

Figure 1.3: Example of an ArUco AR tag. The pattern on these tags are detected by a camera and then used to identify both the tag's pose relative to the camera and the tag's identification number.

individual marker by using a unique pattern of pixels. Within the library, each of the n rows of n pixels are treated as one of n words of n bits. As there are 2^n possible marker configurations, there can be 2^n markers tracked simultaneously in a scene.

Under the methods implemented by Garrido-Jurado et al., each tag is identified via a contour-extraction of an image followed by a polygonal approximation of all contours. As all AR tags are rectangular, any contour remaining after the polygonal approximation is probabilistically a tag. The pose of these contours is then determined via a homography-based projection removal [19], thus resulting in the position and orientation of the tag relative to the camera.

A critical factor that determines the effectiveness of any camera-based tracking system is its ability to detect and track all necessary objects in the

scene in a way that does not hinder the performance of any system dependent upon the real-time tracking information. The developers of ArUco determined via controlled testing that the average processing time of their library to track multiple AR tags was 11.08ms per image, well below the maximum rate needed to display a 60 frame-per-second augmented video.

External Tracking

External tracking methods rely on a dedicated device to locate objects within a scene instead of bundling the tracking responsibility into the camera. There are a number of advantages to choosing this method over the camera-based method which vary depending on which form of tracking device is used.

The invention of x-rays first prompted the interest in developing a method to localize structures in 3D space. However, it wasn't for another 50 years that imaging technology reached a point where this would be achievable with the adoption of MRI and CT imaging in the medical field.

The first widely adopted tracking systems (referred to as "trackers" from here on) were optical trackers [20], which utilize image sensors to triangulate the position of targets. These devices generally have a high accuracy and wide operational area. Earlier systems generally were comprised of charged-coupled device (CCD) cameras, Infrared (IR) light-emitting diodes (LEDs), all integrated into a single platform. There are three main classifications of optical tracking systems:

Videometric systems identify marker patterns in a video feed, generally from a calibrated camera. The camera-based tracking method detailed in this work adheres to this form of optical tracking, although the fact

that the camera that performs the tracking is the same camera that records the displayed augmented scene invalidates it from being an "external" tracking system. Videometric systems are used to detect the positions of crash dummies in car tests [20].

IR trackers implement an optical band-pass filter to remove all ambient light other than Infrared light, simplifying marker identification. IR trackers can be either active or passive in their behavior, with active systems employing stationary CCD units tracking markers that emit Infrared light via attached LEDs. A firing sequence is used to indicate the orientation of each individual marker by illuminating the LEDs in a specific order. By maintaining that the markers adhere to a specific geometric configuration, each marker's pose and identity can be easily tracked.

Compared to their active counterparts, passive IR optical trackers change the locations of the LEDs so that the emitters are affixed to the camera. Reflective spheres are attached to the markers instead, which reflect the light emitted and thus are tracked in a similar fashion to the markers in an active optical system. The chief advantage of a passive system over an active one is the removal of the need to power each individual marker, as the IR light source is in a single centralized location. The work presented in Chapter 3 employs this type of tracking system, specifically the NDI Polaris Spectra system.

Laser tracking systems incorporate a photosensor array coupled with multiple sweeping laser emitters to detect the surface of an object. These systems are not widely used in the medical field.

Although they are highly accurate, there are some drawbacks to optical systems. Optical trackers are generally wired devices, causing clutter in the environment in which they are employed. This issue has been addressed by the development of a number of wireless optical trackers, and the industry standard (according to [20]) Polaris tracking system offers the option of both wired and wireless modes. The largest drawback to optical trackers, however, is the constraint that line of sight must be maintained between the tracker and the object being tracked. In an operating room, where the entire space must be available to clinicians if needed, this is a significant limitation.

To rectify this shortcoming, electromagnetic tracking systems were introduced. These systems incorporate a magnetic field of known geometry to detect and localize small coils or electromagnetic field sensors. Electromagnetic trackers can also be divided into three classifications:

AC-driven trackers were the first form of electromagnetic tracking systems developed. Driven by alternating current (AC), these devices are constructed from three coils positioned in a Cartesian coordinate system that emit three dipole fields. Such systems typically operate in the 8-14kHz range for frequency. Induced voltage in small coils is used to measure the flux of the magnetic field.

DC-drive trackers utilize direct current (DC) instead of AC. These are fairly similar to their AC counterparts, but with the advantage of being able to avoid interference due to eddy currents caused by proximity to conductive metals, which can distort readings from the sensors.

Passive/transponder systems use magnets or transponders to track the position of an object. These are the most recently introduced form of electromagnetic trackers.

Certain other theoretical tracking systems, such as inertial tracking using accelerometers or gyroscopes, have been implemented but have not seen widespread use due to complications in the implementations or general difficulty of use.

1.1.5 Alignment and Registration Techniques

One of the most significant aspects of any augmented-reality system is the process of determining how to align the virtual data in such a way that it meaningfully augments the real world view. Any significant error in the system will be immediately apparent to the user as they realize that the virtual data does not align with the real world display it portrays. To this end, there have been many efforts to determine the best ways of minimizing error when aligning or registering different data sets. To achieve a fully augmented environment, a specific combination of these techniques must be utilized.

Pivot Calibration Pointer tools provide great utility to medical AR systems. Not only are they able to act as a proxy for a surgical device such as a scalpel or probe during development, but they also allow the recording of real-world points (homologous to landmarks in the virtual world used for registration) that would otherwise be indeterminable for lack of a proper method of recording them. However, tracking systems generally will not (or cannot) record the exact tip of the pointer, and instead track a marker

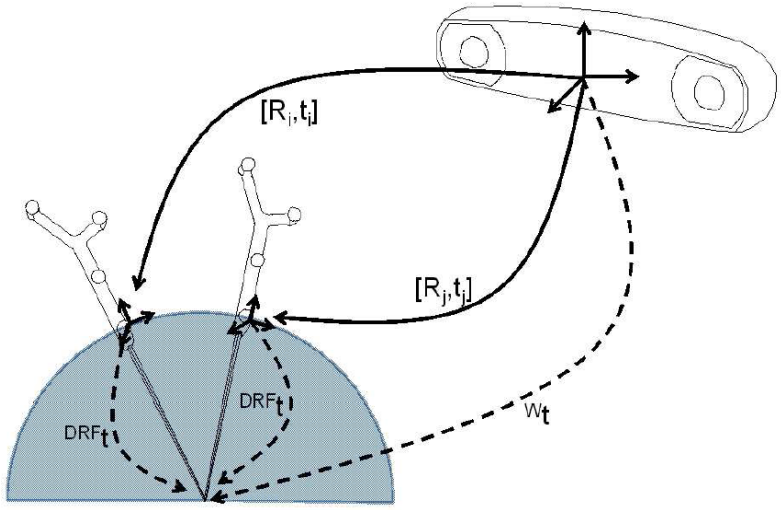

Figure 1.4: Illustration of the pivot calibration process with corresponding translations. ${}^{W}\vec{t}$ represents the transform from the world/tracker origin to the tip of the pointer. Image courtesy of Yaniv et al. [21].

attached to the pointer tool. It is therefore necessary to localize the tip of the pointer with respect to this attached marker via a process referred to as Pivot Calibration.

[21] defines Pivot Calibration as an estimation problem phrased as the following:"Given a set of rigid transformations $[R_i, \vec{t_i}]_{i=1..m}$ obtained by pivoting a tracked object around a fixed world point, estimate the translation ${}^{DRF}\vec{t}$ from the Dynamic Reference Frame (DRF) origin to the pivoting point". A visualization of this scenario is shown in Figure 1.4.

World Registration Once the pointer tool is calibrated it can be used to record landmarks in the real world. This approach allows collection of data that can be used for registration of the virtual coordinate system to that of the real-world tracker. There are a few different ways this can be achieved. The paired-point method is one of the most common, as it follows a fairly simplistic model. Given two measured sets of 3D points with uncertainty

from two different Cartesian coordinate systems, one can solve for the transform between these two sets using a least-squares approximation. There are some variations upon the form of the least-squares solution, but otherwise the algorithm is fairly straight forward.

An alternative method for world registration is the use of surfaces to register two spaces. In this case, one or both of the surfaces are in reality a set of points [20]. One of the earliest implementations of this form of algorithm was the "head-and-hat" algorithm, which aligned segmented surface images of the brain. The term "head-and-hat" refers to the two different surface inputs, with the lower-resolution data — a set of points referred to as the hat and the higher-resolution data — a set of stacked 2D contours referred to as the head. This algorithm calculates the centroid of the head and then minimizes the sum of distances of the rays between the hat points and head centroid and the head surface. This algorithm is considered to be the predecessor of the Iterative Closest Point (ICP) set of algorithms. ICP describes any algorithm that approaches fitting from an iterative corresponding point framework instead of the closest point methods detailed in the "head-and-hat" method.

The Perspective-n-Point Problem: When the tracking responsibilities within the AR system are offloaded to an external tracking system from the camera, the procedure for augmenting a scene must be modified accordingly. When the camera is used for tracking, all tracked objects' positions can be calculated with reference to the camera. In a system where there is an external tracking system, the Image Plane and the tracker space must be registered in some way.

While the camera-based tracking method did not rely on localizing the camera in the world space, the external-tracking workflow described in this work uses localization-based tracking. This method relies on correlating image data and world data recorded using the external system to identify the camera's position in the world frame. The difficulty of this approach is that a camera records two-dimensional images, while it must be localized to a three-dimensional space. For this purpose, there is a method referred to as a Perspective-n-Point (PnP) solution that is able to determine the camera pose in three dimensional space [22, 23, 24]. The term originates from the use of n correspondences among 3D points and the 2D projections of those points. There are many fields in which this method is utilized, including robotics, computer vision, and of course augmented reality [22].

The PnP problem can be solved via two different approaches — iterative or non-iterative. Iterative methods are more reliable when the data is known to be noisy or $n \not\leq 5$, while noniterative methods are generally more efficient in terms of processing time [22]. For the purposes of simplicity, here we only seek to explore one non-iterative algorithm, specifically Li et al.'s [22] RPnP solver. The smallest set that can be used to solve this problem is four 3D/2D point/projection pairs. Using what Li et al. calls the "2-Point Constraint" [22] (1.1), which constrains the unknown depths x_i and x_j between the camera and two of the recorded 3D points P_i and P_j, results in an undetermined system. The term θ_{ij} refers the the viewing angle between P_i and P_j's respective corresponding image points p_i and p_j, and the term d_{ij} refers to the distance between P_i and P_j.

$$x_i^2 + x_j^2 - 2x_i x_j \cos\theta_{ij} - d_{ij}^2 = 0 \tag{1.1}$$

A set of size $n = 3$ results in what Li et al. calls the "Three Point Constraint", which is an application of the 2-Point Constraint to the three possible subsets of two points from given reference points P_i, P_j, and P_k resulting in depths d_ij, d_ik, and d_kj. This system can then be reduced to a single polynomial (1.2) which is passed to a P3P solver. The depths of these points can be solved, but the rotational pose of the camera is not able to be determined using this method.

$$\begin{cases} x_i^2 + x_j^2 - 2x_i x_j \cos\theta_{ij} - d_{ij}^2 = 0 \\ x_i^2 + x_k^2 - 2x_i x_k \cos\theta_{ik} - d_{ik}^2 = 0 \\ x_k^2 + x_j^2 - 2x_k x_j \cos\theta_{kj} - d_{kj}^2 = 0 \end{cases} \qquad ax^4 + bx^3 + cx^2 + dx + e = 0 \tag{1.2}$$

The RPnP solver was created to rectify this issue. It requires a minimum of four 3D/2D point pairs and determines the camera pose relative to the frame in which the 3D points were recorded. The algorithm divides the data set of size n into $(n-2)$ subsets, each of which contains three points and then uses these points to generate a fourth-order polynomial via the Three Point Constraint method (1.2), thus yielding a system of $(n-2)$ fourth-order polynomials, that are used to determine the z-axis of the target rotational matrix. The algorithm then uses SVD to solve for the remaining two rotational elements and the translation vector $\vec{t}$ [22]. In testing, RPnP outperformed multiple other non-iterative methods such as EPnP [25], LHM [26], and DLT [27] in terms of both accuracy and efficiency and is therefore an optimal example of a solution to the PnP problem.

Calculation of Registration Error

In [21], Yaniv describes the ideal registration algorithm as one that is fast, accurate, robust, automatic, and reliable [21]. This means that there are five categories of metrics that can be used to assess the quality of a registration algorithm; Execution time, breakdown point, degree of automation, accuracy, and reliability are all properties that can be used to determine the effectiveness of a registration process.

Of these qualities, three are directly observable - execution time, breakdown point, and degree of automation. Execution time simply details the time it takes for the registration to be completed, from the beginning of the initialization phase to the final output. The breakdown point refers to the algorithm's robustness against outliers within the data. This attribute does not necessarily dictate the algorithm's effectiveness so much as it determines whether or not an outlier rejection phase must be implemented during the initialization stage. Finally, automation is simply a binary value that indicates whether a registration process requires human input or is fully autonomous in its execution.

The two remaining properties - accuracy and reliability - cannot be directly observed and must be calculated using developed methods. Reliability of a specific registration method can be evaluated empirically by using datasets that have been determined to have a "gold standard" transformation. That is, such datasets have been used in the evaluation of multiple algorithms by over 20 research groups [20]. The algorithm can then be tested for reliability by initializing all parameters to those of the gold standard set and analyzing the results relative to the expected transformation from that set. Finally, accuracy is considered to be the most critical measure

of a registration algorithm's quality. Unlike other qualities used to gauge the effectiveness of registration, accuracy varies based upon the data being fit. Therefore, different algorithms for error determination may need to be used dependent upon the type of data used in the work.

1.1.6 Goals of This Work

The scope of the work presented in this paper includes defining workflows for two methods of augmenting a video scene with virtual medical data. The first method - a webcam-tracked AR system - is designed to address the needs of diagnosticians and clinicians who do not have the ability to purchase expensive tracking equipment. This technique utilizes a standard computer and a commercially available webcam to facilitate augmentation, relying on the webcam to track the real scene. Since this method requires few components that cannot easily be acquired or are generally ubiquitous in modern society, it is inexpensive to implement (The system used in this paper cost roughly $40 ignoring the cost of the computer).

The second method explored in this work is an augmented reality system that uses an external tracking system, specifically the Polaris Spectra, to track the scene and facilitate augmentation. The goal of this portion of the work is to attempt to show that the webcam technique can be improved upon for use within a more sophisticated surgical environment. This method focuses less on reducing cost and more on reducing error.

The expected contributions from this work therefore are an inexpensive AR system that can be easily configured and used and a more costly system that can improve the performance of the process without any significant alterations to the components or configuration process.

1.1.7 Thesis Outline

In this paper we describe the workflow to generate an augmented reality environment using a simple camera as both a real time imagine device, as well as an inexpensive and sufficiently accurate tracking device (Chapter 2), alongside a virtual model of a physical phantom extracted from a Computed Tomography (CT) image dataset using off-the-shelf segmentation and surface rendering tools.

We also present a modified form of this workflow that utilizes a high-accuracy medical tracking system as an alternative to the webcam-based tracking (Chapter 3). As a result, the camera view in both of these applications is augmented with a virtual pose of the model that is rendered according to the position and orientation of the camera, resulting in an intrinsically registered virtual view overlaid onto the real camera view.

This approach enables users to visualize structures located beneath the surface which cannot be seen using a simple camera. By transposing this application to minimally invasive interventional guidance, this technique enables "beneath-skin" visualization of organs and tissues that require treatment and which cannot be visualized using an endoscope or laparoscope, but can be rendered as virtual models extracted from high resolution data Computed Tomography (CT) or Magnetic Resonance (MR) imaging data.

We believe that these approaches will open up new horizons for combining real and computer-generated information for several applications that require virtual exploration of beneath-surface structures that otherwise cannot be seen in a non-invasive manner. A direct application is in virtual anatomy training and simulation an educational tool that can be employed to expose non-medical staff to views of the underlying anatomy as reconstructed

from life-size medical imaging datasets and registered to mannequins that enable the user to maintain the direct relationship between the real and virtual world. In addition, a second application consists of the development of platforms for surgery and therapy simulation for clinical staff, as an introduction to minimally invasive approaches prior to their implementation in cadaveric, animal and human studies.

Chapter 2

An Intrinsic Camera-Based Video-Enhanced Augmented Reality Approach

2.1 Introduction

Although there are multiple ways of tracking and augmenting a visual scene, one of the most common approaches is the inherent use of the visualization device — generally a camera — as the tracking system. In addition to simplifying the systems integration necessary to configure the environment, camera-based tracking is inexpensive and therefore accessible to individuals who may not be able to afford expensive tracking systems. Our objective in developing this work is to provide a suitable guideline for configuring such an environment to catalyze future research in methods that enhance these camera-based tracking AR environments.

Our process, developed as a module integrated into the open-source community-supported platform for medical image analysis and visualization — *3D Slicer* — entails several components: intrinsic camera calibration, followed by an extrinsic camera calibration utilizing AR tags attached to each real-world component to determine the camera's world position, registration of

the real world to its virtual model counterparts, and lastly the video augmentation. Note that the real-to-world registration process involved a pivot-based pointer tip calibration process (i.e., a protocol used used to relate the tip of the pointer to its attached AR marker detected by the camera), followed by a paired-point registration that uses homologous landmarks in the real and virtual world to align the two coordinate systems such that the poses of tracked objects could be rendered in the virtual space.

To assess the quality of the resulting model-enhanced video augmentation, several registration statistics were employed including the Fiducial Localization Error (FLE), Fiducial Registration Error (FRE) and Target Registration Error (TRE). As introduced in the previous chapter, the Fiducial Localization Error (FLE) measures the uncertainty associated with the localization of a landmark point, the Fiducial Registration Error (FRE) provides an estimate of the residual error associated with the paired landmarks after registration, while the Target Registration Error (TRE) quantifies the registration error at landmarks not employed during registration.

2.2 Tools and Methods

2.2.1 Overview

There are seven key components to the workflow described in this paper. The first is the medical imaging software platform into which all components were integrated — *3D Slicer* — and serves as the main software platform for the project. Because of restrictions imposed by 3D Slicer, which relies heavily on open-source libraries such as the Visualization Toolkit (VTK), Insight Segmentation and Registration Toolkit (ITK), and OpenCV,

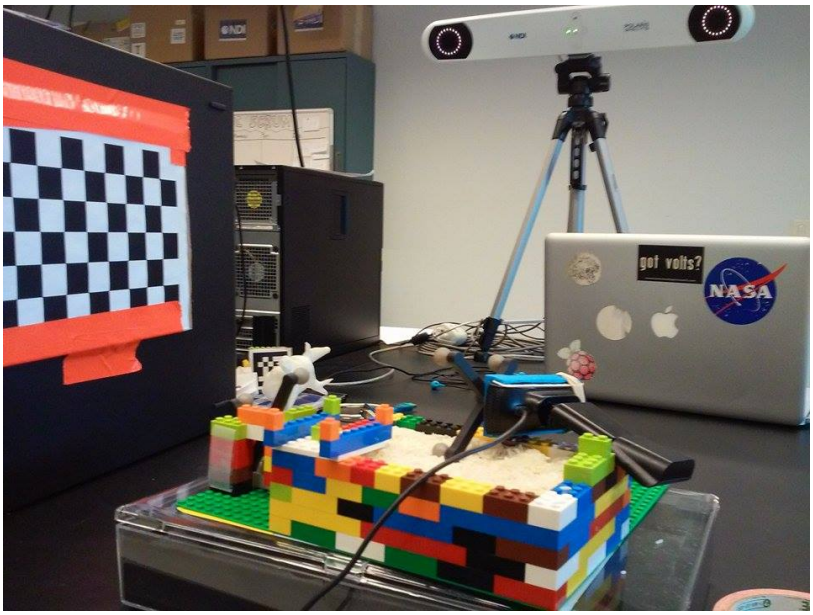

Figure 2.1: This image illustrates all the components utilized to generate the inherent camera-based AR environment.

VTK was used as part of the rendering pipeline. Moreover, given the open-source flavour of 3D Slicer, all implementations were restricted to the C++ and Python languages.

To serve as a physical model or hypothetical patient, a physical model comprised of LEGO blocks was constructed and then scanned using a Computed Tomography (CT) scanner. The use of the LEGO phantom was justified by the accuracy of the LEGO blocks and hence the resulting accuracy of the reconstructed model, which enables us to re-use a previously acquired CT image of the same LEGO model, without the need to re-acquire new images of the physical object.The CT image dataset was segmented using 3D Slicer to create the virtual model of the LEGO phantom.

A USB webcam - specifically the Logitech C920 - was used to acquire a view of the scene. The entire deliverable was implemented using a desktop computer running Windows 7 64-bit.

2.2.2 Platform

The application is implemented in the form of a module in the 3D Slicer open-source platform architecture and uses readily-available libraries from the VTK and OpenCV. In addition, the Open-IGT Link protocol enables communication between 3D Slicer, OpenCV, and the camera, the latter serving as both a real time imaging device, as well as a tracking device. The modular design of the platform facilitates the addition of new components and features. The platform also supports visualization of pre-operative MR, CT and other modalities simultaneous to the camera feed, in addition to tracked surgical tools and virtual models extracted from the above-mentioned imaging modalities. Moreover, the platform also provides the ability to selectively combine the different imaging components, set transparency levels for overlays, visualize volumetric data from orthogonal or oblique planes, as well as generate dynamic cinematic sequences.

One of the key features of 3D Slicer that prompted its use in this project was the ability of the platform to track multiple frame transformations in real-time, providing data management and class structures for storing and interacting with the transforms. This saved valuable amounts of time in the development cycle, as having to devise a system of organizing and updating transforms. Additionally, many tools were available on the platform that provided key functionalities necessary for this workflow to operate, such as the pivot calibration and fiducial registration modules, discussed later in this chapter.

2.2.3 Environment Workflow

This application could be implemented by simultaneously tracking the camera, as well as the real surgical scene using an external tracking system, such as an optical or electromagnetic spatial localization system typically used during computer-assisted surgery. To streamline the focus of this application and reduce the cost, the system was implemented such that both the real world visualization and tracking were performed a simple web camera and AR tags rigidly attached to the phantoms and to a pointer device used to perform the world registration i.e. establish the correspondence between the real and virtual world. The following steps were necessary in order to augment the video view of the real world with its virtual counterpart:

Step 1 - Rigidly affix reference frames (via AR tags) to each object of interest in the scene.

Step 2 - Establish a relationship between the pointer tip and the AR tag depicted by the camera (Pointer calibration).

Step 3 - Establish a relationship between the real world phantom and its virtual counterpart (VTK model extracted from a CT scan of the real phantom) using point-based registration (World calibration).

Step 4 - Establish a relationship between the video image of the real world and its corresponding real world scene (Intrinsic and extrinsic calibration).

2.2.4 Intrinsic and Extrinsic Camera Calibration

Theory

The intrinsic and extrinsic matrices are two components of what is known as the "camera calibration procedure". The product of the intrinsic and extrinsic matrices describes the mapping between a three-dimensional world point $\vec{X}^W$ and the corresponding two-dimensional image point $\vec{x}^{im}$, that is a result of the projection of $\vec{X}^W$ onto the image plane of the camera. The intrinsic parameters are dependent upon the pinhole camera model (**Fig.** 2.2), which represents a camera through a simplistic geometrical model that reduces the necessary calculations for calibration.

Throughout this paper, a 2D point on the image plane will be denoted by $\vec{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$, while a 3D point in the world space will be denoted as $\vec{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$. Any homogeneous transform between two three-dimensional coordinate systems will be identified using the format ${}^{S}T_{D}$, where superscript $\mathbf{S}$ is the originating coordinate system and subscript $\mathbf{D}$ is the target coordinate system. *e.g.* the transform ${}^{\mathbf{global}}\mathbf{T}_{\mathbf{local}}$ in the equation $\vec{x^{local}} = {}^{\mathbf{global}}\mathbf{T}_{\mathbf{local}}\vec{x^{global}}$ would transform point $\vec{x^{global}}$ in the global frame to $\vec{x^{local}}$ in the local frame. As the inverse of any homogeneous transform from one coordinate system to another is representative of the inverse relationship between the two coordinate systems, the inverse of a homogeneous transform such as ${}^{S}T_{D}$ will thus be denoted either in the form $({}^{S}T_{D})^{-1}$ or in the form

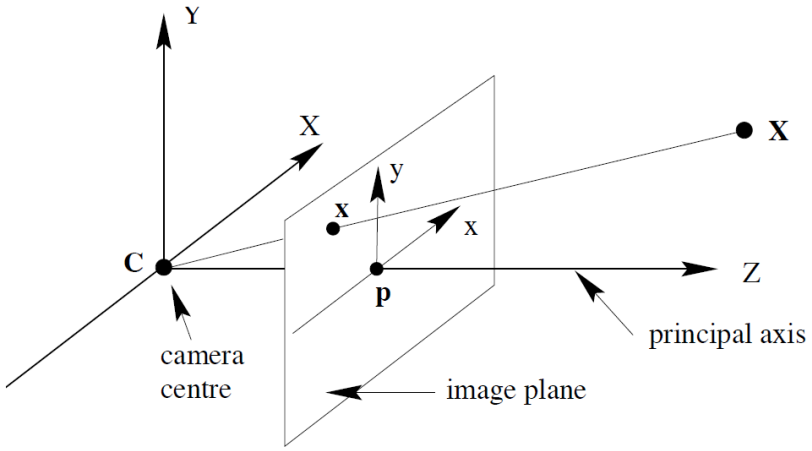

Figure 2.2: Illustrative diagram of the pinhole camera model, a purely geometric representation of a camera. Image courtesy of Rhody et al. [28]

.

${}^{D}T_{S}$.

As shown in **Fig.** 2.2, world point $\vec{X}^{W}$ sits along a ray passing from the camera center through image point $\vec{x}^{im}$. This means that, given $\vec{X}^{W} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$, we can extrapolate that $\vec{x}^{im} = \begin{bmatrix} f_x * x/Z \\ f_y * y/Z \\ 1 \end{bmatrix} = \begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix}$.

Camera projection includes three different coordinate systems in its calculations. The first is the Image Plane, a two-dimensional coordinate system containing all image points $\vec{x}^{im}$. The Camera Coordinate System (CCS) is a three-dimensional system that originates at the camera center, with the three axes defined by the camera's orientation. Finally, the World Coordinate System (WCS) is the global reference frame through which all objects in the scene are described. All of these planes are associated through the camera projection equation shown in **Eq.** 2.1.

$$\mathbf{P} = \mathbf{S}^{\mathbf{CCS}}\mathbf{T}^{\mathbf{W}}_{\mathbf{cam}}\mathbf{T}_{\mathbf{CCS}} \tag{2.1}$$

The derivation of this equation is comprised of four steps. First, the image plane is associated to an intermediate plane referred to as the Focal Plane. The relationship between these two planes is defined by a translation and rescaling transform based upon the camera's lens properties (**Eq.** 2.2) [28]. This matrix **S** transforms an image point $\vec{x}^{im}$ to point $\vec{x}^{cam}$ on the focal plane. m_x and m_y refer to the scale factors for each axis from metric units (e.g. mm) to pixels. x_0 and y_0 represent a translation of the plane origin from the camera's principal point to the corner of the image. s is a skew factor. For this procedure we assume $s = 0$.

$$\mathbf{S} = \begin{bmatrix} m_x & s & x_0 \\ 0 & m_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

As the focal plane acts as an intermediary between the image plane and the CCS, and as the CCS is a three-dimensional plane while the image plane is two-dimensional, the transformation ${}^{\mathbf{CCS}}\mathbf{T}_{\mathbf{cam}}$ between $\vec{x}^{cam}$ on the focal plane and the point $\vec{X}^{CCS}$ on the CCS is a 3x3 transformation matrix shown in **Eq.** 2.3 [28], appended with a column of zeros to account for the projective nature of the transform between the focal plane and the CCS.

$$\vec{x}^{cam} = \left[{}^{\mathbf{CCS}}\mathbf{T}_{\mathbf{cam}}|\mathbf{0}\right]\vec{X}^{CCS} \tag{2.3}$$

Combining **Eq.** 2.2 and **Eq.** 2.3 results in the Intrinsic Matrix **K** (**Eq.** 2.4) (appended with a column of zeros to account for change in dimensionality), which describes the full relationship between the image plane and the

CCS.

$$\vec{x}^{im} = \Big[\mathbf{K}|\mathbf{0}\Big]\, \vec{X}^{CCS} = \mathbf{S}^{\mathbf{CCS}}\mathbf{T}_{\mathbf{cam}}\vec{X}^{CCS}$$
$$\mathbf{K} = \begin{bmatrix} m_x & s & x_0 \\ 0 & m_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} fm_x & fs & x_0 \\ 0 & fm_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

Having established the relationship between the image plane and the CCS, it is now necessary to calculate the transformation ${}^{\mathbf{W}}\mathbf{T}_{\mathbf{CCS}}$ between the CCS and the WCS. This transformation constitutes the Extrinsic Matrix (**Eq.** 2.5).

$$\vec{X}^{CCS} = {}^{\mathbf{W}}\mathbf{T}_{\mathbf{CCS}}\vec{X}^{W} \tag{2.5}$$

The combination of the extrinsic and intrinsic matrices results in the camera's projection matrix, which fully describes the projection of $\vec{X}^{W}$ to $\vec{x}^{im}$ (**Eq.** 2.6). By calculating this matrix, it is possible to fully augment a real scene with virtual data by simply setting the pose of each virtual component to that of its real-world counterpart. However, due to the fact that a projection from a two-dimensional plane to a three-dimensional plane results in an under-determined system, the equation shown in **Eq.** 2.6 cannot be manipulated to solve for $\mathbf{P}$. In the case of the webcam-based tracking system, an external library was used to remedy this issue. The calculations for the two- to three-dimensional projection can be viewed in **Eq.** 3.2.4.

$$\vec{x}^{im} = \mathbf{S}\vec{x}^{cam} = \mathbf{S}^{\mathbf{CCS}}\mathbf{T}_{\mathbf{cam}}\vec{X}^{CCS} = \mathbf{S}^{\mathbf{CCS}}\mathbf{T}_{\mathbf{cam}}({}^{\mathbf{W}}\mathbf{T}_{\mathbf{CCS}})\vec{X}^{W}$$

$$\vec{x}^{im} = \mathbf{P}\vec{X}^{W} \tag{2.6}$$

$$\mathbf{P} = \mathbf{S}^{\mathbf{CCS}}\mathbf{T}_{\mathbf{cam}}({}^{\mathbf{W}}\mathbf{T}_{\mathbf{CCS}})$$

The ArUco Library: To construct a referential frame for interpreting the real world coordinates and orientation of the target object, AR tags were used in conjunction with the ArUco tag recognition library. Markers were attached to the LEGO phantom and the pointer tool. Because the camera is used as the tracking system, it is assumed to act as the WCS. This means that the camera origin is centered at the world origin and the CCS and WCS are one and the same. To construct a referential frame for interpreting the real world coordinates and orientation of the target objects, these AR markers were used in conjunction with the ArUco tag recognition library [19].

The ArUco library was created with the goal of creating a technology that could efficiently track multiple markers efficiently by using a stochastic approach instead of a "complete evaluation of the search space". The library stores m markers made up of a $(n+2) * (n+2)$ pixelated black-and-white grids (Seen in **Fig.** 1.3). An outside border two pixels wide aids detection by isolating the inner nxn grid. This inner grid is used to identify each individual marker by using a unique pattern of pixels. Within the library, each of the n rows of n pixels are treated as one of n words of n bits. As there are 2^n possible marker configurations, there can be 2^n markers tracked simultaneously in a scene.

To track these markers, ArUco analyzes each video frame individually (**Fig.** 2.3). First, images are converted to gray-scale and segmented using

a local adaptive threshold to identify the prominent contours in the scene (**Fig.** 2.3, image (b). After this, contours are extracted using the Suzuki-Abe method [29] (**Fig.** 2.3, image (b)), the results of which are passed to a polygonal approximation using the Douglas-Peucker algorithm [30] (**Fig.** 2.3, image (d)). By discarding any contour not approximated to a polygon with four vertices (all markers are rectangular), the remaining contours are probabilistically those of the AR markers. The code from these markers is then extracted by first eliminating the effects of perspective projection through homography calculations and divided into a grid (**Fig.** 2.3, image (e)). Finally, each marker is checked against an internal dictionary containing all valid marker IDs. Any marker that shares an ID with a dictionary entry is tracked and its homogeneous transform is stored for access by the user.

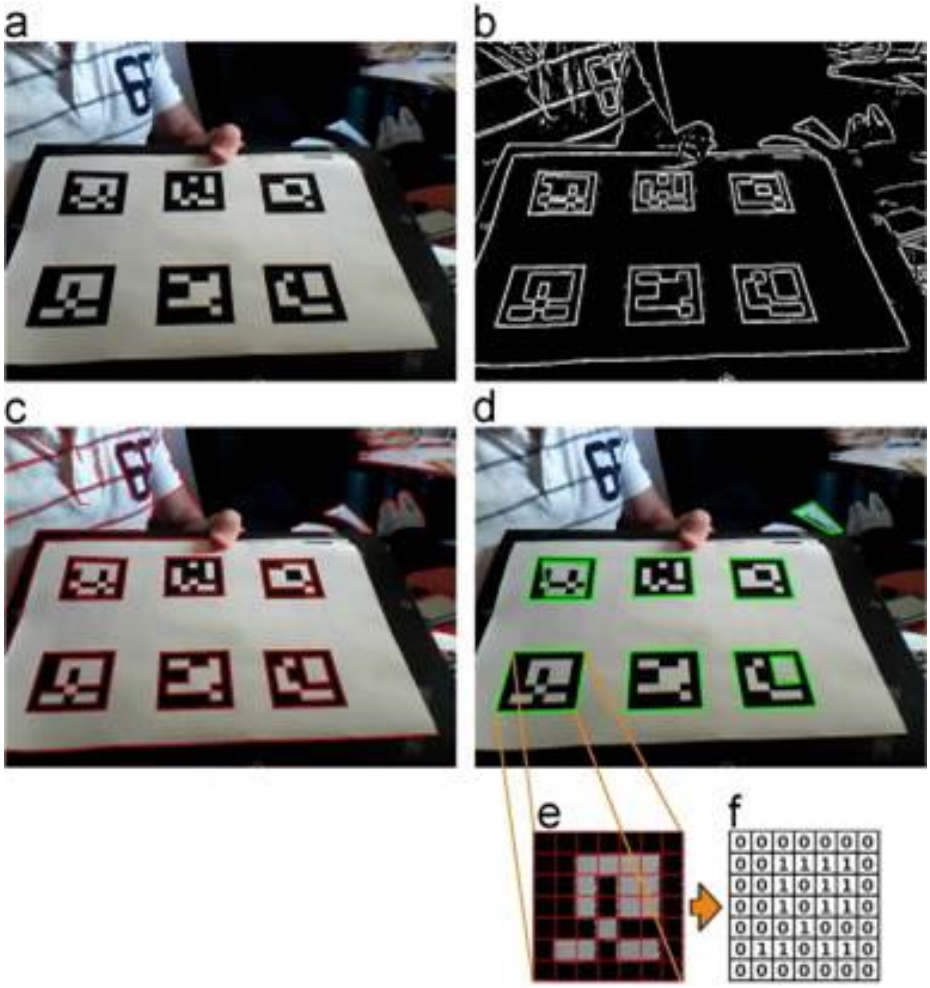

Figure 2.3: Each step of the ArUco tag detection and tracking process. (a) shows the image captured before processing; (b) is a contour extraction of the image; (c) represents the results of a contour detection; (d) shows the polygonal approximation and color removal of each marker; (e) visualizes the results of a perspective transformation of the marker in (d); (f) demonstrates the bit assignments used to create the tag's identification number. Image courtesy of Garrido-Jurado et al. [19].

Implementation

Using this tool, the extrinsic camera calibration matrix ${}^{\mathbf{W}}\mathbf{T}_{\mathbf{CCS}}$ - a homogeneous transformation matrix which describes the camera's position relative to a given AR tag - was obtained. Because of the constraints put in place by VTK's implementation of virtual cameras, elements of this matrix were extracted and applied to the scene camera independently. First, the translational elements of the extrinsic matrix were used directly to set the position of the scene camera in the render space. Then the first column of the 3x3 rotation matrix R (i.e., a sub-matrix of the 4x4 extrinsic calibration matrix ${}^{\mathbf{W}}\mathbf{T}_{\mathbf{CCS}}$) was used to set the view-up vector associated with the virtual camera - a vector orthogonal to the perspective vector that dictates the perceived vertical direction as viewed by the camera. This process positioned the virtual camera in such a way that positions and orientations of both the real and virtual cameras were identical within their respective spaces.

No unit conversions were necessary in this implementation, as both ArUco and VTK employ a millimeter scale for all spatial transformations and coordinate frames. However, when using cameras, there is a distortion present in the images recorded due to the lenses necessary for the camera operation. This distortion is represented as a barrel distortion on the image that causes bowing of the acquired images outward from the image center. If not accounted for, correct alignment of a virtual model to an image is impossible as ArUco will be unable to correctly determine the pose of AR markers.

To un-distort the camera image, we employed the intrinsic camera calibration procedure proposed by Zhang et al. [31], in which multiple views of a checkerboard with known dimensionality and uniformity (**Fig.** 2.4) were

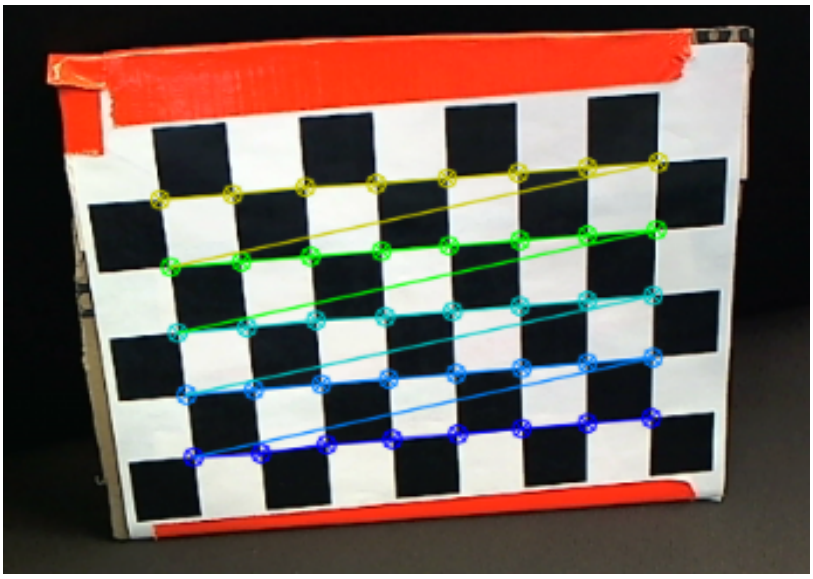

Figure 2.4: Camera calibration checkerboard used to correct for lens distortion overlaid with OpenCV recognition pattern. The overlaid lines are applied by OpenCV for identification purposes. The board in this image was affixed to a rigid piece of cardboard to ensure minimal warping would occur as any bends or creases in the paper could affect detection or registration performance.

acquired. These images, along with the webcam's extrinsic calibration matrix, were then passed to an OpenCV utility to generate the required matrix necessary to un-distort the acquired images.

2.2.5 World Calibration

Once the real and virtual cameras were aligned, the real and virtual representations of the viewed scene needed to be registered. Given the proposed system is envisioned for use as a visualization tool in the context of minimally invasive therapy, and compounded by the to obtain the coordinates of individual landmarks corresponding to the real-world model, a second, dynamic A R tag was introduced into the system, which would be attached to a surgical instrument, therefore enabling real-time tracking of the tool

in the same coordinate system as the rest of the scene. The extrinsic matrix returned by this tag was programmatically inverted so that the resulting transform would be associated with the position and orientation of the tool, rather than the camera.

Due to the displacement between the tip of the pointer tool and the tracked marker attached to it, a pointer calibration (hereby referred to as a Pivot Calibration) was necessary to establish a relationship between the two. Because the pointer tip is for all intents and purposes a single point, this transformation is a pure translation from the origin of $\mathbf{T_{point}}$ to the pointer tip $\vec{p_{point}}$. For this process, we used 3D Slicer's pivot calibration module (Bundled with the IGT extension package), which simplified the process of calculating the optimal transform. This module took as an input the transform relating to the pointer marker. A data acquisition process was then run, during which the pointer tool was pivoted on its tip with the attached marker clearly visible to the camera. The module recorded the pose of the marker over this time period and then calculated and returned the optimal transformation ${}^{\mathbf{P}}\mathbf{T}_{\mathbf{T}}$ as well as the RMS error. The estimated static pointer calibration transform was then applied to the probe transform (i.e. the inverted tool extrinsic matrix), which was updated according to the new position and orientation of the tool with respect to the camera.

Once the static transform was applied to the pointer tool's frame, a rigid-body fiducial-based registration was performed to establish a spatial correspondence between the virtual model and its corresponding real counterpart. A series of fiducial markers (i.e., landmarks) were selected on the virtual model; these same features were then recorded as a homologous, yet separate fiducial set from the real-world model using the camera-tracked pointer.

Note that it is important to select landmarks whose locations can be easily and accurately identified in both the real and virtual models, otherwise the spatial correspondence between the two worlds will be compromised. The points recorded in this fashion appeared as markers in the virtual space, albeit in the wrong position and orientation. The virtual-world marker dataset was then registered to the real-world marker set using the Fiducial Registration module built into 3D Slicer; the module uses two sets of markers of the same dimension (N points, each characterized by their inherent x-, y- and z-coordinates) and returns a transformation matrix, accompanied by a metric that quantifies the registration error, commonly referred to as the Fiducial Registration Error (FRE). Lastly, the returned transformation between the real and virtual world, often referred to as the world calibration transform was then applied to all the components of the virtual world, resulting in a correct alignment between the real-world video feed and the virtual world.

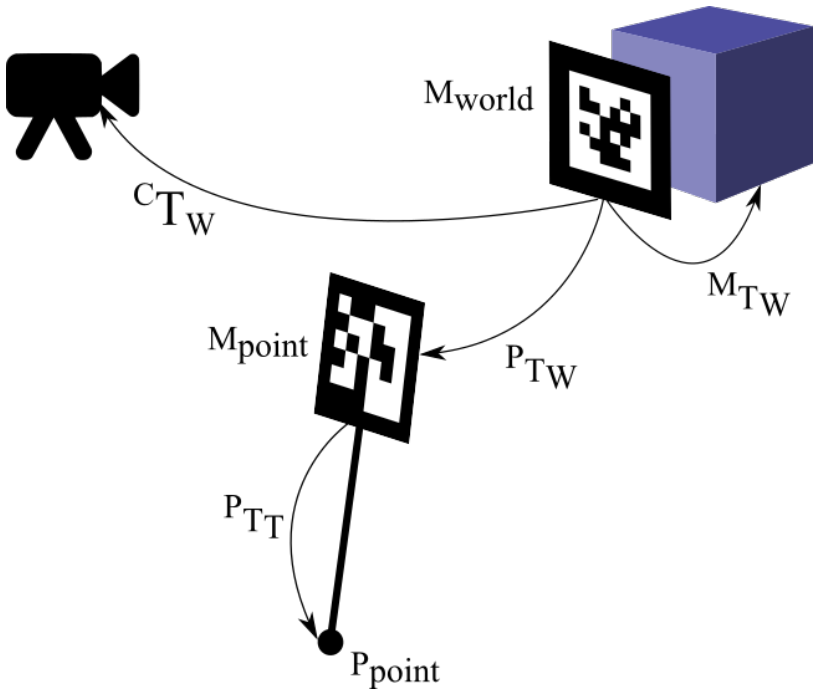

Figure 2.5: This graphic represents the entire physical scene, comprised of camera, phantom (represented as blue cube), phantom marker, pointer, and pointer marker.

2.2.6 Implementation on Intervention-Mimicking Phantoms

Following the intrinsic and extrinsic camera calibration, the technique was implemented and demonstrated using a LEGO phantom and ongoing studies are being conducted on a patient-specific left atrium mimicking phantom. The IGT LEGO phantom was first designed and disseminated by Yaniv et al. [32] and constitutes a very accurate, yet highly inexpensive device that allows for the development, implementation, testing and validation of newly formulated tools, devices, surgical techniques, and user-dependent surgical skills in a laboratory setting, while emulating a somewhat simplified clinical environment in which all variables can be controlled in an efficient and organized fashion. Moreover, thanks to the precision of the LEGO blocks,

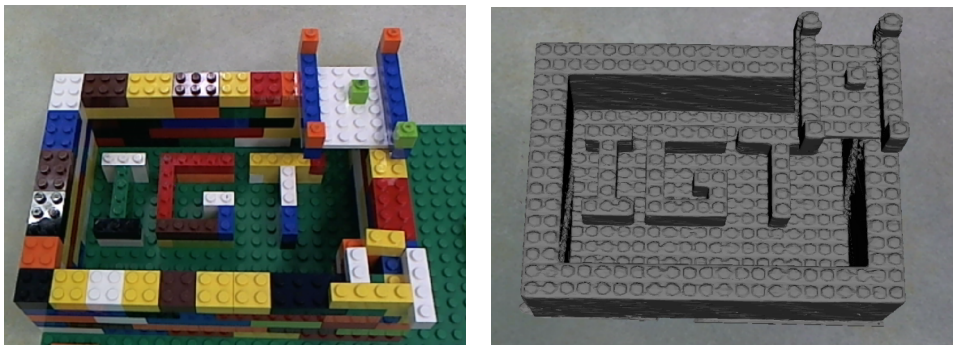

Figure 2.6: Image-Guided Therapy (IGT) LEGO phantom used for in-laboratory validation of new tools and techniques for computer-assisted navigation and visualization (left panel) accompanied by its virtual counterpart generated via segmentation from a CT scan of the phantom (right panel).

exact replicas of phantoms can be rebuilt and used alongside virtual representations obtained by automatically segmenting previously acquired CT images of the device, without the need to re-acquire CT scans of a newly built phantom (**Fig.** 2.6).

2.2.7 Error Correction and Handling

While the system behaved moderately well as previously described, there were notable issues with jitter in the model due to AR tag detection. It is hypothesized that the majority of these issues were caused by the slight defects in the surface of the tags, which must be as flat as possible to obtain the best results. To combat these issues, we implemented a combined method for improved image tracking stabilization and error reduction.

The proposed and implemented approach consists of a variable-width averaging window applied to the returned extrinsic transforms from the AR tags. The width corresponds to the number of frames over which the transforms are averaged. This window provided the benefit of smooth motion

of the virtual model, but resulted in a width-dependent delay in the motion of the pointer to the correct position as the average weighted the position to previous frames. Nevertheless, this issue could be further remedied by employing a weighted average that assigns a higher weighing to the recent frames than to the older ones.

In addition to the averaging window, an outlier detection feature was also added. This feature allows the user to calibrate the system for a set duration, during which the noise distribution in the static reference AR tag is recorded and a standard deviation of the noise is computed. Once this calibration step is performed, the user has the option to enable noise removal, by automatically discarding any transform that contained values which deviated by less than 0.05 standard deviations for the static transform or 0.01 standard deviations for the probe transform.

2.3 Results

The first assessment concerned the alignment of the world coordinates to those of the virtual environment. The accuracy of this alignment was dictated by the RMS error of four different alignment phases: Pivot Calibration Error (PCE), Fiducial Localization Error (FLE), FRE, and Target Registration Error (TRE):

- The PCE refers to the error in determining the transformation between the pointer tip and the AR tag attached the the pointer.

- The FLE refers to the error in locating a single point in space (measured by recording 60 points at the same real-world location and measuring their spread in the virtual space).

- The FRE refers to the error when aligning a set of real-world points to a corresponding set of virtual points, and measured the spatial correspondence estimated based on all the landmarks used to perform the world calibration.

- The TRE refers to the error achieved when mapping real-world points to the virtual space after registration has been performed. Of the three error metrics, TRE is the most critical as it ultimately determines the usefulness of the image guidance environment the user is employing.

These error metrics were quantified using different hardware components and are summarized in **Table** 2.1. The entire registration process was repeated for the webcam tracker three times with a recording of the respective metric made at each step of the process.

Iteration	PCE (mm)	FLE (mm)	FRE (mm)	TRE (mm)
1	.928	.639	2.12	5.08
2	1.15	.890	2.98	4.44
3	0.963	0.752	2.21	3.21

Table 2.1: Results from three different iterations of the registration process.

The webcam FLE and FRE are higher than the corresponding errors using higher-quality hardware and an alternative software input mechanism the OpenIGTLink protocol used to pass information from the tracking systems to 3D Slicer (See **Chapter** 3 for full results of said alternate systems). The Logitech webcam performed worse, but this was anticipated. While the NDI units have advantages such as larger fields of view, no lighting restrictions, and no line-of-sight requirements (in the case of the electromagnetic unit only), the webcam still performed quite well overall when factoring in

the cost-to-performance ratio.

While their TRE associated with the webcam employed was much greater than the Polaris Spectra, the Polaris webcam costs two orders of magnitude less than the NDI Polaris Spectra unit. The cost-to-accuracy ratio for the hardware/software system we have developed is therefore extremely competitive, once again, for non-medical applications that also do not require the use of regulator-approved technology.

Also in the context of the intended application in simulation, teaching and training, the use of the camera for tracking also limits the technology required to generate an augmented reality visualization environment. Had a separate spatial localization system been employed, the camera would have needed to be tracked extrinsically, therefore adding an extra layer of hardware, software and transforms to the current application.

When compared to the RMS error values of the same tests on tested systems (said systems being the Polaris Spectra infrared imager and the Aurora magnetic imager), the results were deemed acceptable when consideration was given to the unavoidable error present in the imaging algorithms needed for detecting the AR Tags.

Our software further seeks to align the webcam video feed with our tracked virtual environment in real time. An example of this overlaid alignment is shown in **Fig.** 2.7. The error in this alignment was quantified by capturing screen captures of the overlaid images and manually measuring the offset between the image and the overlaid model using corresponding landmark datasets. This process is demonstrated in **Fig.** 2.8 and the recorded errors are summarized in **Table** 2.2. This process did not require prohibitive computational power, as 3D Slicer was able to maintain a refresh rate in

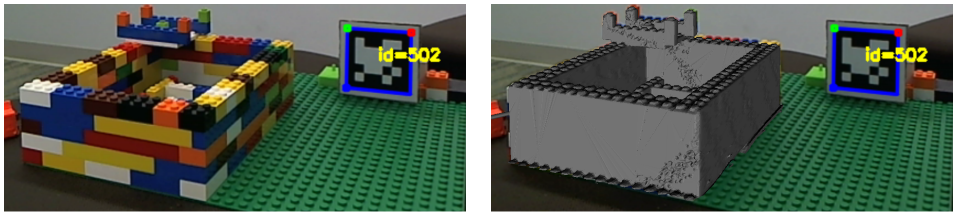

Figure 2.7: Raw video feed of the IGT LEGO phantom (left panel) accompanied by the virtual model overlaid onto the real-world view (right panel).

excess of 40 frames per second while processing the overlay.

As demonstrated in **Fig.** 2.7, the overlay process is capable of near-perfect alignment in some instances. Unfortunately, while the overlay tends to be very close to the true image when the algorithm starts, moving the camera around the object can introduce error into the alignment.

To assess the quality of the image overlay system we used the spatial alignment of the real and virtual features in terms of their alignment error i.e., how well do the virtual surface views of the model align with the real surface view under quasi-static conditions (i.e., no sudden motion of the camera with respect to the images scene). This error was calculated using an image space fiducial registration process. First, a two-dimensional projection of the rendered scene from the camera feed was recorded and marked with fiducial points at key features on the object. A second set of fiducials was then recorded by matching homologous features on the projection of the virtually-rendered model. These two sets of fiducials sets were then passed into a TRE algorithm, which, in turn, returned the alignment error between the 2D real-world view and the overlaid virtual model view as a RMS TRE metric. An example of this fiducial placement process is given in **Fig.** 2.8.

The alignment quality, hereby referred to as Video Registration Error (VRE), was quantified across the real-world and virtual fiducials using the

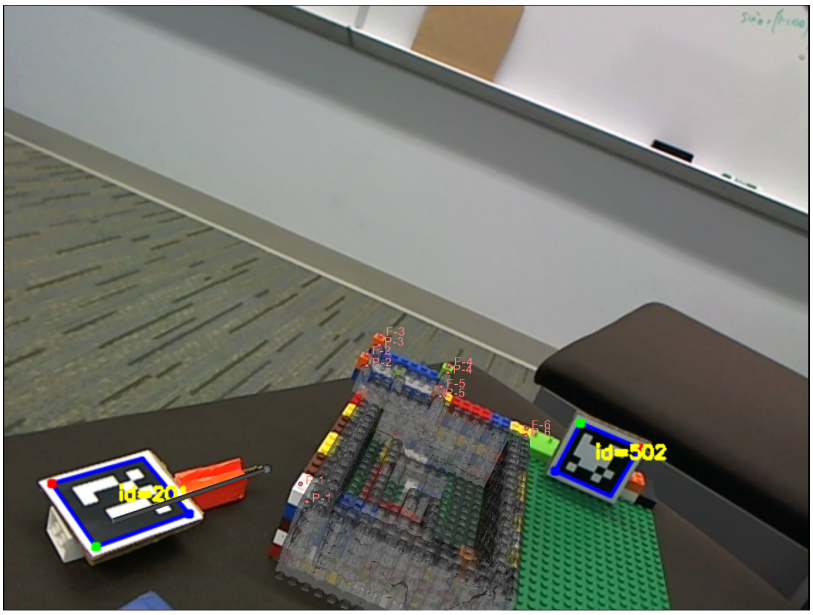

Figure 2.8: Video alignment error was computed by comparing the offset between corresponding point sets (denoted with prefixes F- and P- in the image).

Tracking System	Lowest VRE (RMS, mm)	Mean VRE (RMS, mm)	Highest VRE (RMS, mm)
Webcam Tracker	4.57	9.51	17.72

Table 2.2: As computed from multiple viewing angles for each camera. The lowest and highest errors represent the best and worst case alignments detected while moving the camera through the possible orientation space.

RMS TRE as an assessment metric.

The VRE varied significantly based on camera quality and orientation. Best, worst, and average cases for each camera are reported in **Table** 2.2.

The best video overlays corresponded to the scenario in which the camera was in the same orientation relative to the AR tag as it had been during the initial fiducial registration process. Tracking became significantly impaired when viewing the static AR tag head-on.

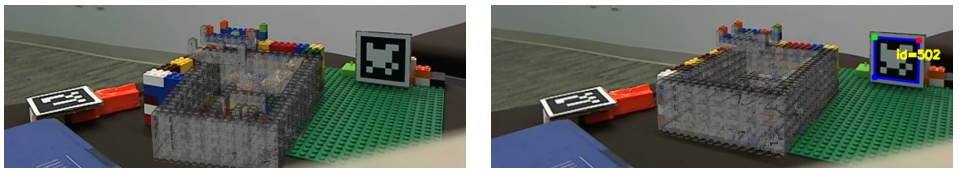

Figure 2.9: View of scene before distortion correction (left) and after (right).

2.4 Discussion

2.4.1 Error

One significant source of error which our software eliminates is the error due to camera distortion. Camera lenses by their nature distort images by bending incoming light to focus it onto the sensor array. By accounting for this distortion in incoming images we achieved significantly improved tracking fidelity, as shown in **Fig.** 2.9.

Despite accounting for camera distortion parameters, the error in video alignment is still greater than one would expect given the relatively small TRE of the systems. One possible explanation for this error is the somewhat unreliable nature of the AR tag detection library employed. The system is highly sensitive to camera focus and lighting conditions, factors that may interfere with the tag detection, which, despite reasonable accuracy leading to low TRE, led to sub-optimal accurate estimates of the world coordinates relative to the camera.

The lack of fidelity of tag detection was more apparent when viewing the static reference tag head-on, which is a well-known problem in computer vision. In this orientation the system had a very difficult time aligning the model to the video because the ArUco software could not generate an accurate estimation for the depth of the tag in space. This shortfall could

potentially be addressed in the future by using of a board consisting of several static reference tags as opposed to a single reference tag currently employed. The multiple reference tags detected concurrently should allow for a more accurate consensus on estimating the camera pose with respect to the viewed scene. Alternatively, the simultaneous use of a system of cameras could allow for similarly improved results, while also permitting the user to view the augmented reality environment in 3D. Higher quality lighting conditions and camera hardware may also make the tags easier to detect, thereby helping to reduce video alignment error.

2.4.2 Advantages and Limitations

The potential benefits of the AR approach can be easily seen when the target area in the real world is visually occluded. Consider a scenario depicted in **Fig.** 2.10, in which the LEGO structure is filled with rice, preventing the visualization of the internal structures – the IGT lettering inside the phantom. If one were to try to "operate" on the "T" for instance, it would be difficult to do so without hitting other structures along the way. Using the augmented reality view, nevertheless, it is easy to bring the pointer directly to the target location. By translating this analogy to a real, surgical scenario where target locations are occluded by skin and/or tissue, as the organs to be treated are accessed in a minimally invasive fashion through a small incision, without providing ample access to the internal organ our proposed augmented reality solution would permit surgeons to insert the instruments and navigate them to the target with sufficient accuracy, despite the lack of direct, real visualization of the scene available through small incisions.

The most significant advantage of inherent camera-based tracking and

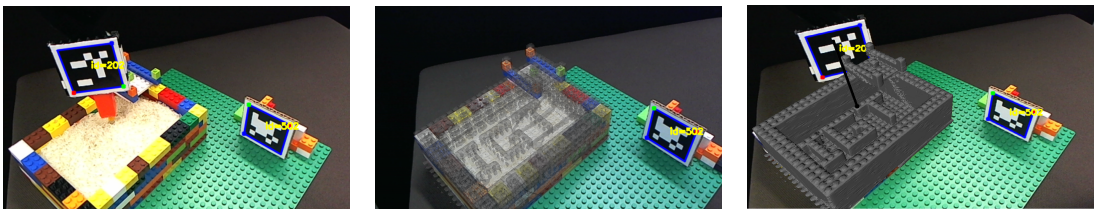

Figure 2.10: Model with targets occluded (left panel) alongside video overlay (middle panel), enabling augmented reality-based image-guided navigation of the tracked pointer to the desired targets (right panel).

AR visualization is the significantly reduced cost of the equipment compared to other external tracking systems, making virtual tool tracking feasible for most audiences, yet limited to applications that focus on simulation, teaching and training, rather than the actual performance of image-guided interventions that require superior accuracy and use of specialized equipment. In addition, this approach also limits the components required to build the apparatus, as the camera is used for real world view acquisition and tracking, with no need for external localization devices.

One limitation of the current implementation is the slight inaccuracies associated with the camera tracking, especially along the depth direction. One approach to be considered to improve these inaccuracies involves the use of a pair of cameras, which would enable better triangulation for depth perception, also enabling stereoscopic visualization. Another limitation is the difference between the focal length of the camera and that of the rendered which will require an additional calibration step or better positioning of the phantom with respect to the camera to optimize distance from focus.

2.5 Conclusion

We believe the described simple and cost-effective augmented reality platform will provide new opportunities for combining real and computer-generated information for several applications that require virtual exploration of occluded structures that otherwise cannot be seen in a non-invasive manner.

A direct application is in virtual anatomy training and simulation an educational tool that can be employed to expose high school students and personnel with no medical training to views of the underlying anatomy as reconstructed from life-size medical imaging datasets and registered to mannequins that enable the user to maintain the direct relationship between the real and virtual world.

A second application consists of the development of platforms for surgery and therapy simulation for clinical staff, as an introduction to minimally invasive approaches prior to their implementation in cadaveric, animal, and human studies. Because of the minimal cost due to low-cost equipment and reliance on open-source software, the system is ideal for physician training in communities that may not have the resources for high-end medical equipment.

In its current stage the application uses a single camera source and the display is rendered on a traditional monitor. In the next chapter we will discuss a reimplementation of this system using an external tracking system.

Chapter 3

An External Tracking-Based Video-Enhanced Augmented Reality Approach

3.1 Introduction

As we demonstrated in **Chapter** 2, it is possible to rely entirely on the camera for establishing a spatial reference frame and use the camera as a tracking device. However, this technique is limited to the inherent uncertainty of the camera and any geometric and illumination distortions it features, as well as the ability to use AR tags affixed to the objects and instruments to be tracked as a means for their position ans orientation to be determined by the camera.

One approach to address these limitations is to employ external tracking systems to identify the position and orientation of all instruments (i.e., camera, pointer, physical space etc) that comprise the experimental or practical apparatus. The work described in this chapter seeks to implement an augmented reality application that makes use of an external tracking system to track the camera and other components in lieu of the camera-based tracking method presented and implemented in Chapter 1. This addition to existing methods allows for added support of the Aurora magnetic tracker

and Polaris Spectra optical tracker. Introducing these systems into the tracking environment has the potential to increase tracking accuracy, minimize or eliminate the issue of obstacles interfering with tracking (especially for magnetic tracking), and provide more versatility to the system within a surgical environment. The effectiveness of these systems was analyzed using the same methods used to analyze the camera-based tracking.

3.2 Tools and Methods

3.2.1 List of Resources Used

The implementation of an externally tracked AR system is to a certain extent similar to the procedure of using the camera-based tracking. The only difference in the external tracking scenario is the addition of an external tracking device and the markers the device tracks. In the case of this work, the system was implemented using a Polaris Spectra stereo optical tracker and three markers with reflective infrared spheres: one marker was attached to the camera, one was attached to the pointer and the third marker was attached to the "surgical scene" imaged by the camera that was augmented with its virtual model. As the camera itself must be tracked when an external tracking system is utilized, a specialized mount was designed using SolidWorks (a CAD program) and 3D printed. A marker was attached to this mount, which was then rigidly affixed to the camera. The full set of additional pieces of equipment used is shown in **Fig.** 3.1.

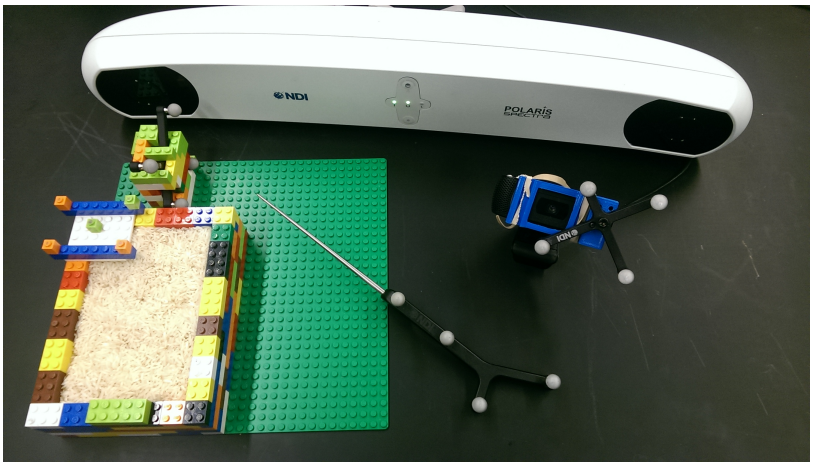

Figure 3.1: Collection of all components (excluding desktop computer) used in this system.

3.2.2 Platform

The 3D Slicer platform was once again chosen as the platform upon which the implementation of this work would be developed. Much of the rendering process was extracted from the work presented in **Chapter** 2, although the external tracking process was implemented in a separate module with minimal dependency upon the camera-based tracking module.

3.2.3 Tracking

While the camera-based application was focused on creating a system that could effectively render an Augmented Reality scene at a low cost and was simple to configure, the focus of the externally-tracked system was to achieve increased accuracy and increase robustness by allowing visualization of objects that may not be immediately visible to the camera due to occlusion or obfuscation. The steps of the procedure (listed below) do not vary from the alternate implementation, but the process by which each step was performed varied slightly from the steps listed in **Chapter** 2.

Step 1 - Rigidly affix reference frames (via IR markers) to each object of interest in the scene, including the camera.

Step 2 - Establish a relationship between the pointer tip and the attached tracking sensor (Pointer calibration).

Step 3 - Establish a relationship between the real world phantom and its virtual counterpart (VTK model extracted from a CT scan of the real phantom) using point-based registration (World calibration).

Step 4 - Establish a relationship between the camera position and its tracked marker (Intrinsic and extrinsic calibration), thus associating the camera to the tracked scene.

3.2.4 Intrinsic and Extrinsic Calibration

Theory

The largest difference in the workflow for an externally-tracked AR application from a camera-based tracking method is the process of extrinsic calibration. In a system where the camera is used for tracking objects in the scene the CCS and the WCS are equivalent - that is, a homogeneous transformation between the two would be equivalent to identity matrix $\mathbf{I}$. However, in an externally-tracked system this is no longer necessarily the case. As shown in **Fig.** 3.2, there is an extra transform ${}^{\mathbf{W}}\mathbf{T}_{\mathbf{CCS}}$ necessary to relate the WCS to the CCS.

In the previous system, much of the necessary calculations for obtaining the extrinsic transform were offloaded to the ArUco library, which used image data exclusively to calculate the position of markers in the world. In the case of the external tracking system, the WCS was no longer dictated

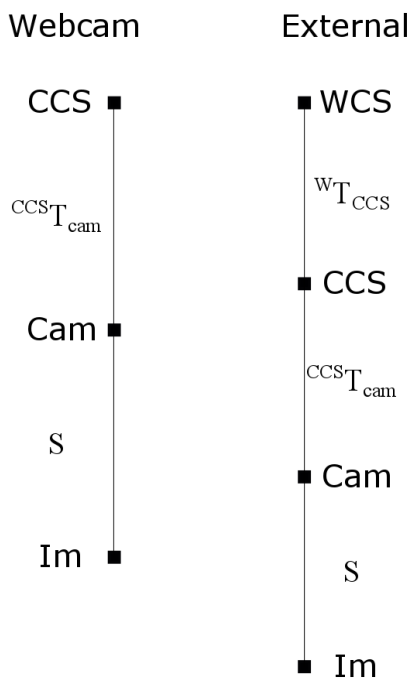

Figure 3.2: Spatial comparison of the two tracking methods. Note the extra transform ${}^{W}T_{CCS}$ that much be calculated between the WCS and the CCS in the externally-tracked case.

by the camera pose, instead being centered at the origin of the tracker's coordinate system (**Fig.** 3.3). As such, and due to the fact that tracking was no longer being performed by a camera, the position of the camera center in the scene (i.e. the Extrinsic Matrix matrix) was no longer able to be determined using the previous methods.

At first glance it may seem that the solution to this issue should have been a simple one. By manipulating the camera projection equation as shown in **Eq.** 2.6, one could solve for the extrinsic matrix ${}^{\mathbf{W}}\mathbf{T}_{\mathbf{CCS}}$ (Shown as the product $({}^{C}T_{M}){}^{W}T_{M}$ in **Fig.** 3.3) using linear algebra. However, due to the loss of data inherent in a three- to two-dimensional projection and

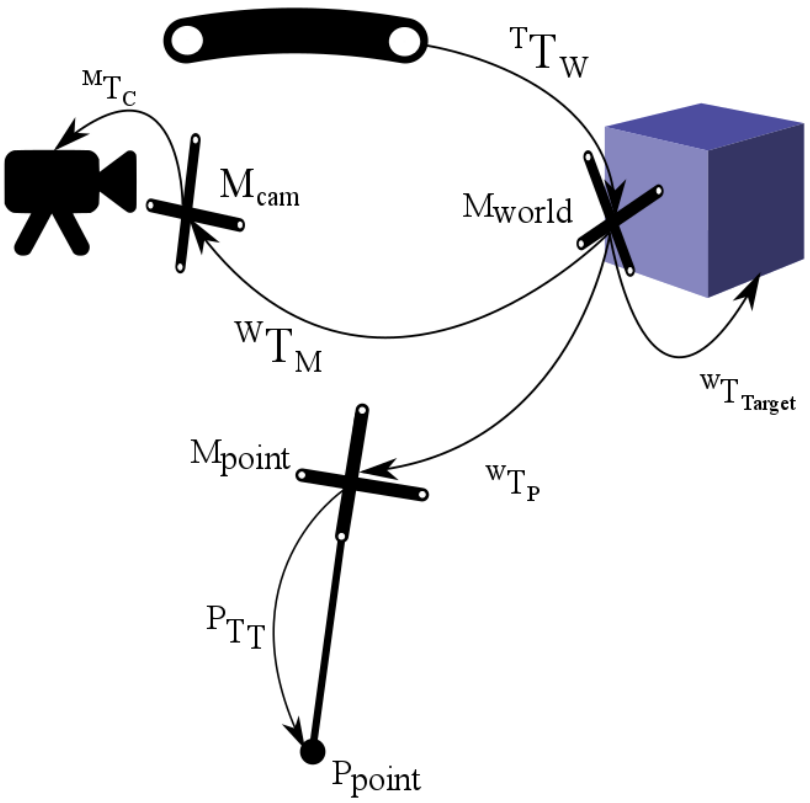

Figure 3.3: This graphic represents the entire physical scene, comprised of camera, phantom (represented as blue cube), phantom marker, pointer, and pointer marker.

the resulting non-invertibility of the modified 3x4 intrinsic matrix $\left[\mathbf{K}|\mathbf{0}\right]$ presented in **Eq.** 2.4, the resulting system of equations is not solvable using traditional algebraic methods **Eq.** 3.1. Even after reducing the matrix to Euler angles, the system remains undetermined at six variables represented in three equations.

$$\vec{x}^{im} = \Big[\mathbf{K}|\mathbf{0}\Big]\ {}^{\mathbf{W}}\mathbf{T}_{\mathbf{CCS}}\vec{X}^{CCS}$$

$$\begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix} = \begin{bmatrix} fm_x & fs & x_0 & 0 \\ 0 & fm_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{aligned} x_{im} = t_z x_0 + x(R_{31}x_0 + fm_x R_{11} + fR_{21}s) + y(R_{32}x_0 + fm_x R_{12} + fR_{22}s) + \\ z(R_{33}x_0 + fm_x R_{13} + fR_{23}s) + fm_x t_x + fst_y \\ y_{im} = x(R_{31}y_0 + fm_y R_{21}) + y(R_{32}y_0 + fm_y R_{22}) + \\ z(R_{33}y_0 + fm_y R_{23}) + t_z y_0 + fm_y t_y \\ 1 = t_z + R_{31}x + R_{32}y + R_{33}z \end{aligned} \tag{3.1}$$

There was, however, a way to circumvent this computational challenge. By recording multiple 2D/3D point pairs while keeping ${}^{\mathbf{W}}\mathbf{T}_{\mathbf{CCS}}$ static, it was possible to create an overdetermined system by representing the vectors $\vec{x}^{im}$ and $\vec{X}^{CCS}$ in equation **Eq.** 3.1 as 3xn and 4xn point sets respectively, where n represents the number of points recorded. While this method allowed for an overdetermined system, it also introduced some error into the system through uncertainty in the recording process.

The SolvePnP Function

As a multivariable system with uncertainty in any of its variables cannot be solved using traditional algebraic means, it was necessary to implement a regression approach to solve this system. For this specific issue in which a

set of 2D camera points and 3D world points are given with the desire to calculate the camera pose, the problem is referred to as a Perspective-n-Point (PnP) problem. This problem states that the camera pose can be estimated using a minimum of four 2D/3D point sets. There are multiple ways to solve this regression ([22], [23], [24]), but for this specific implementation we utilized Zhang's method [33], which was implemented via the *SolvePnP* function in the *OpenCV* library. This function accepts as input a set of n 2D image points, n 3D points in the CCS, and the intrinsic matrix and distortion coefficients found using Zhang's camera calibration method [31] and calculates the optimal extrinsic matrix for the camera using a least-squares regression.

The 2D and 3D points were recorded using a 7 x 9 checkerboard. The image coordinates of each of the checkerboard's corners were recorded programmatically using *OpenCV*'s *findChessboardCorners* function. The 3D points were recorded using the calibrated pointer tool with respect to the IR marker attached to the camera. In this workflow, the values recorded using the pointer tool are given with reference to the world marker. However, as *SolvePnP* returns a transform describing the extrinsic relationship of the camera origin to the world frame, the extrinsic matrix returned by *SolvePnP* was equivalent to $({}^{M}T_{C})^{W}T_{M}$. As the extrinsic matrix ${}^{C}T_{M}$ desired was one relative to the attached IR marker $\mathbf{M}_{cam}$, it was necessary to transform all recorded points by ${}^{W}T_{M}$ to move them into the IR marker space to obtain the correct transform.

Implementation

Using this tool, the extrinsic matrix ${}^{\mathbf{W}}\mathbf{T}_{\mathbf{CCS}}$ — a homogeneous transformation which describes the camera's position relative to world marker $\mathbf{M}_{\mathbf{world}}$ — was obtained. The resulting transform was used to set the pose of the virtual camera. As was done previously in **Chapter** 2, elements of this matrix were extracted and applied to the scene camera independently. First, the translational elements of the extrinsic matrix were used directly to set the position of the scene camera in the render space. Then the first column of the 3x3 rotation matrix $\mathbf{R}$ (i.e., the upper right-hand sub-matrix of ${}^{\mathbf{M}}\mathbf{T}_{\mathbf{C}}$) was used to set the view-up vector associated with the virtual camera - a vector orthogonal to the perspective vector that dictates the perceived "vertical" direction as viewed by the camera. This process positioned the virtual camera in such a way that positions and orientations of both the real and virtual cameras were identical within their respective spaces.

No unit conversions were necessary in this implementation, as both the Polaris Spectra and VTK employ a millimeter scale for all spatial transformations and coordinate frames. Unlike the camera-centric implementation shown in **Chapter** 2, removal of the lens distortion present in the camera is not critical to a successful augmentation. This is likely due to the offloading of tracking to the optical tracker instead of the camera, as a distorted camera image would doubly affect the tracking accuracy and the display accuracy, whereas an external tracking system would only see an effect in the display accuracy. This does not mean a distortion correction is not recommended, as doing so still increases augmentation accuracy, but a successful augmentation is still possible without this step.

Implementation on Intervention-Mimicking Phantoms

Initially the same model used in **Chapter** 2 was used to test the procedures detailed. However, as the objective of implementation of the external tracking system was to ensure the highest possible accuracy, and as there was slight visual distortion of the LEGO model, a similar model was created using CAD software to the exact measurements of the physical structure. This model was then registered using the same process described in **Eq.** 2.2.5. The quality of this registration was then compared to that of the original model using the returned RMS error from the Fiducial Registration module. The CAD model showed a registration error of .74 mm whereas the original segmented model resulted in an error of .98 mm. As such, the CAD model was used for the duration of this procedure.

Interpolation of Checkerboard Points

Being that the solution for the PnP problem is not algebraic and instead based upon sampled data, it is more effective to utilize a greater number of sampled 2D/3D point pairs to reduce the chances that the regression gets stuck in a local minimum and returns an incorrect extrinsic matrix. Therefore, it was necessary to implement a technique to allow a large number of data points to be collected with relative ease.

The chosen method was to find the intermediate chessboard corners by recording the four outermost corners of the board and interpolate the inner points using the chessboard plane's basis vectors. The user first records the four outer corners $\vec{p}_{1...4}$ of the chessboard. These corners are then used to derive the basis vectors of the board space $\vec{u}_1$ and $\vec{u}_2$ (**Eq.** 3.2).

$$
\begin{aligned}
\vec{p}_n &= \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \; n \in ([1,4] \cap \mathbb{Z}) \\
\vec{u}_1 &= \frac{\vec{p}_2 - \vec{p}_1}{||\vec{p}_2 - \vec{p}_1||} \\
&= \frac{\begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix}}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \\
\vec{u}_2 &= \frac{\vec{p}_3 - \vec{p}_1}{||\vec{p}_3 - \vec{p}_1||} \\
&= \frac{\begin{bmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \end{bmatrix}}{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2}}
\end{aligned}
\tag{3.2}
$$

After this, the basis vectors are used to calculate each intermediate corner p_{ij} where i is the row index of the corner and j is the column index (**Eq.** 3.3). These point p_0 refers to the originating corner from which each intermediate point is calculated. To reduce error propagation due to uncertainty in the four points used to calculate the basis vectors, the value of x_0 within the calculation of a given point $x_i j$ is equivalent to the nearest of the points $\vec{p}_{1...4}$, thus reducing the maximum scaling of any error present within $\vec{u}_1$ and $\vec{u}_2$ by half.

$$
\vec{p}_{ij} = i * \vec{u}_2 + j * \vec{u}_1 \tag{3.3}
$$

Iteration	PCE (mm)	FLE (mm)	FRE (mm)	TRE (mm)
1	0.0789	0.0440	0.660	1.12
2	0.112	0.0473	0.740	1.16
3	0.086	0.0435	0.861	1.20

Table 3.1: Results from three different iterations of the registration process using the Polaris Spectra.

3.3 Results

As with the webcam-based tracking method, the full registration process was completed three times. The results of these registrations can be seen in **Table** 3.1, which demonstrates the accuracy of the Polaris Spectra as a tracking method in terms of the PCE, FLE, TRE, and TRE. When compared to the results in **Chapter** 2, it is clear that the Spectra is superior to the webcam in terms of registration quality, although this does not necessarily dictate a higher augmentation quality as well.

An assessment of the full augmentation process using manually recorded points for the extrinsic calibration was performed similarly to the process demonstrated in **Chapter** 2. After calculating the extrinsic matrix, six fiducial points were selected on the LEGO phantom and were marked in seven different camera views. These same fiducials were then marked on the virtual model of the phantom in images acquired at the same camera poses. Each individual set of 6 point pairs and the entire pointset of 42 point pairs were then passed through a TRE algorithm. The value returned from this algorithm dictated the quality of the augmentation in the recorded images (**Table** 3.2).

To verify that the interpolation worked as intended and did not introduce any significant error into the environment, the three dimensional point sets

Iteration	Lowest VRE (RMS, mm)	Mean VRE (RMS, mm)	Highest VRE (RMS, mm)
1	1.24	4.60	6.87
2	1.49	4.10	6.04
3	1.45	4.25	6.49

Table 3.2: VRE for augmentation using manually recorded 3D points to calculate the extrinsic matrix. The lowest and highest errors represent the best and worst case alignments detected while moving the camera through the possible orientation space.

consisting of 48 points that were recorded during registration were then interpolated using the four corners of the checkerboard (**Fig.** 3.4). Both the recorded and interpolated points from each the three registration iterations were then registered to one-another using 3D Slicer's Fiducial Registration module. The resulting RMS errors, shown in **Table** 3.3 show that the interpolated points have a minimal deviation from the recorded points. It was initially concluded that point interpolation in its current form was detrimental to the performance of the system based upon these results, but in the interest of robust testing procedures the full augmentation process was completed twice — once with recorded points and once with interpolated points. The results of error analysis on these two augmentations is discussed in the following section.

Iteration	RMS Error (mm)
1	0.589
2	0.563
3	0.570

Table 3.3: RMS errors between interpolated and recorded points over three different iterations of the registration process.

The Video Registration Error (VRE) was quantified once again by recording screen captures of the overlaid images and manually measuring the offset between the image and the overlaid model using corresponding landmark

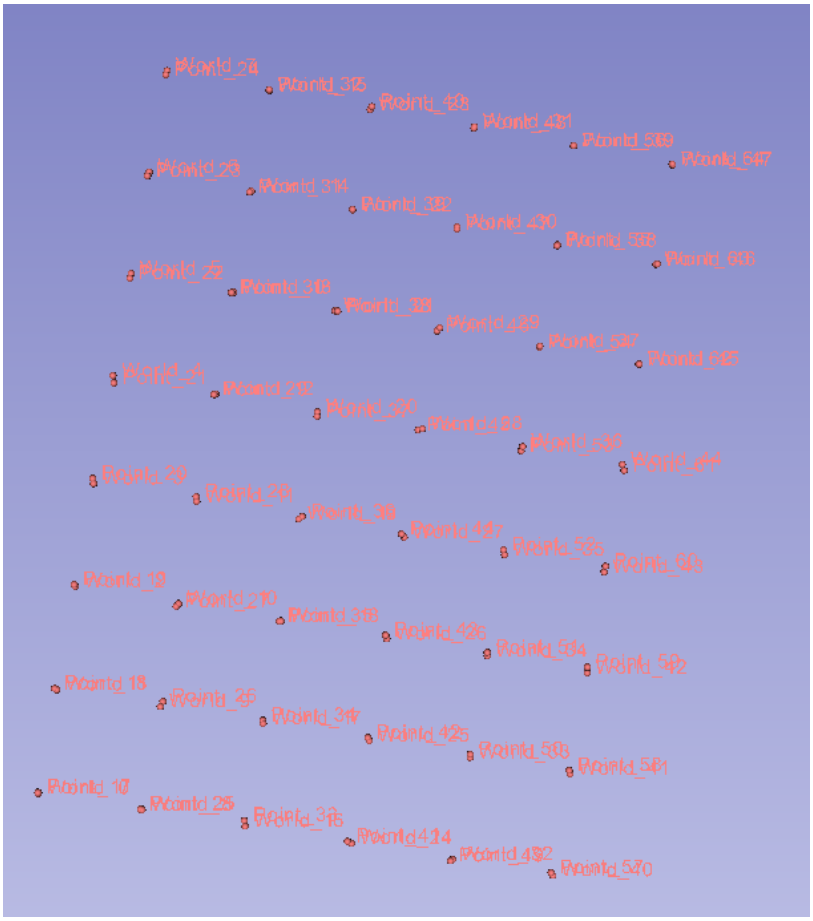

Figure 3.4: Visual scene of the interpolated and recorded point sets in the virtual space. It is clear from this image that the two data sets are fairly similar as there are no large disparaties between corresponding points.

datasets. Unlike the process demonstrated in **Chapter** 2, however, in this case there were two different augmentations to assess, these being the augmentation using the extrinsic matrix calculated via manually recorded points and the augmentation using the extrinsic matrix calculated via interpolated points. These same fiducials were then marked on the virtual model of the phantom in images acquired at the same camera poses on both the recorded and interpolated augmentations. The process is demonstrated in **Fig.** 3.5 and the recorded errors are summarized in **Table** 3.2. This did not require

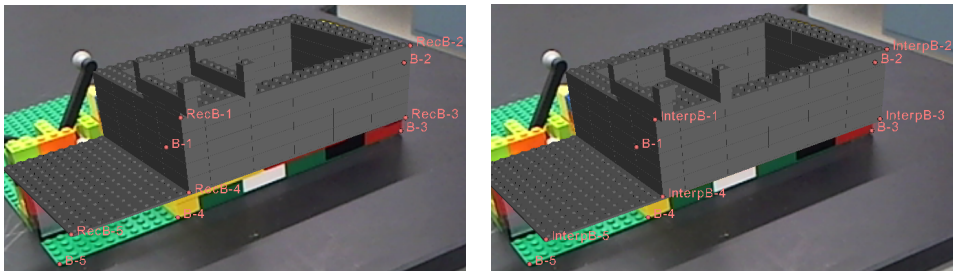

Figure 3.5: Video capture of the error assessment process using the extrinsic matrix calculated via recorded points (left panel) and interpolated points (right panel). The fiducials from the interpolated (Labeled as "InterpB-[x]" in the image) points and from the recorded (Labeled as "RecB-[x]" in the image) points are shown alongside the fiducial points recorded manually from the raw video feed.

Point Set	Lowest VRE (RMS, mm)	Mean VRE (RMS, mm)	Highest VRE (RMS, mm)
Recorded Point Augmentation	1.49	4.10	6.04
Interpolated Point Augmentation	1.39	4.36	6.17

Table 3.4: VRE for augmentation of the same registration using both manually recorded and interpolated 3D points to calculate the extrinsic matrix. The lowest and highest errors represent the best and worst case alignments detected while moving the camera through the possible orientation space.

prohibitive computational power, as 3D Slicer was able to maintain a refresh rate in excess of 25 frames per second while processing the overlay.

The alignment was quantified across the real-world and virtual fiducials using the RMS TRE as an assessment metric. The quality of alignment varied significantly based on camera quality and orientation. Best, worst, and average cases are reported in **Table** 3.4.

3.4 Discussion

3.4.1 Assessment

In some ways the results shown were as anticipated. The externally-tracked system showed an increase in performance over the webcam-based tracker

in terms of both registration and best-case VRE. However, with regard to the average and worst-case performances, the system actually lagged slightly behind the webcam method slightly. Given the accuracy of the registration results shown in **Table** 3.1 this was unexpected. However, the point of failure that resulted in this underperformance can be isolated to the extrinsic calibration process, given that the registration was so accurate for the external system. The exact cause within this step of the augmentation process is not clear. It may be rooted in specific implementation details of the SolvePnP function, as certain elements of the extrinsic matrix output by said function consistently required rescaling - specifically, the second element of the translation column in the extrinsic matrix was scaled by a factor of 10 consistently. It was necessary to thus downscale this value for the augmentation to align correctly. It is problematic as well that SolvePnP is effectively a black box in terms of implementation-specific behavior. Later research should focus on identifying the exact cause of this scaling issue within SolvePnP to determine if it has an impact on performance elsewhere in the results.

With regards to the interpolation quality, there was a slight improvement in VRE when using the interpolated points in the best case, but overall performance was degraded from manually recording. The difference, however, is fairly small. Interestingly enough, the augmentation calibrated using interpolated points actually performed better than the one using manually recorded points in the best case scenario and nearly the same in the worst case. This feature may be useful in the future as a way of sacrificing a small amount of accuracy for a less tedious registration process.

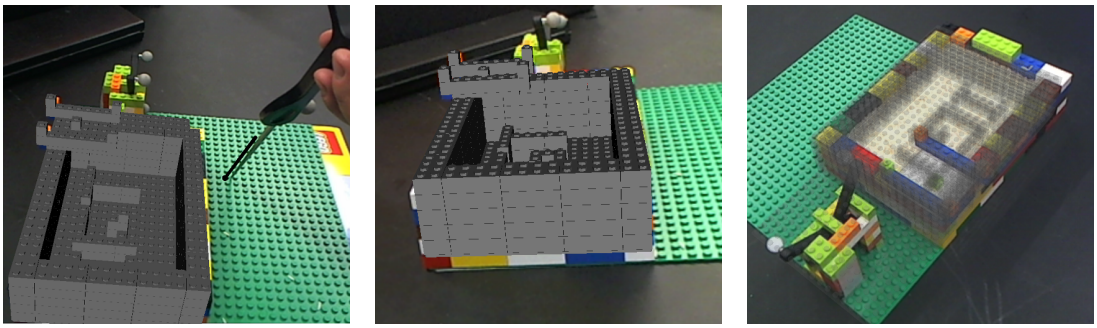

Figure 3.6: Screen captures of the final augmentation when using an external tracking system.

Error Propagation

Once again, for some camera poses, the error in video alignment is greater than anticipated based upon the small TRE of the individual system components. Whereas in the previous chapter we postulated that this larger error could be due to the ArUco library and the variability inherent in using a visual processing algorithm to detect planar surfaces in a three dimensional space, this repeated disparity between anticipated results and actual results forces us to reconsider that hypothesis and instead put forward the estimate that the issue lies either within the rendering pipeline that we have created, and thus is not an issue of theory but of implementation, or more likely that the error is simply due to the propagation of the individual errors within the workflow and is unavoidable given the conditions under which the system was configured, at least given the current lack of understanding about SolvePnP's specific implementation details. A possible workflow adaptation for future work is to recreate this system with a high resolution camera, using a calibration chessboard that has been verified to be printed with sub-millimeter accuracy and is ensured to be flat across the entire surface.

Should this result in a better augmentation, it can be assumed that the error present in our results is due simply to the conditions under which the experiment was conducted. Otherwise, the fault may very well lie within the rendering pipeline or some other component of the system.

3.4.2 Advantages and Limitations

As demonstrated by the accuracy results, the system still performed better than the camera-based counterpart. Not only is the mean VRE smaller, but the spread of the error over multiple camera views is more more consistent, with a range of 10.0 mm RMS for interpolated points vs the camera-based system's 13.2 mm RMS, rendering the external tracking approach as more feasible and more reliable than camera-based tracking.

Additionally, the offloading of tracking responsibilities from the camera capturing the scene view means that the view of the tracking marker can be entirely obscured from view without affecting the augmentation process.

The system is not without limitations, however. The Polaris Spectra utilizes infrared light to track individual markers. This phenomenon implies that any IR saturation in the environment could affect the tracking quality of the system. This was experienced firsthand during the implementation of the system when ambient sunlight from a nearby window completely disabled the tracker's ability to detect markers. However, given that the intended use for this system is within a confined surgical or diagnostic space, this issue should not be significant for the large majority of applications that are not susceptible to IR fluctuations. Obviously there is also the issue of the remaining error present in the system, which is currently assumed to be

due to error propagation due to uncertainty present in the multitude of registration steps performed during system configuration, and, of course, the inherent error of the employed tracking system, which rapidly varies according to where within the the optimal tracking volume the measurements are recorded.

3.5 Conclusion

The described workflow and augmented reality platform builds upon our previous work using camera-based tracking and demonstrates a notable reduction in error between the two systems, which was the primary goal of this work.

This system has applications in all of the same areas as the camera-based system. However, it also has the advantage of being able to be utilized in low-light environments,as long as there are no significant sources of infrared light (as the Polaris Spectra tracker utilizes infrared light to track markers).

There are, of course, several improvements that could be made to the system. Error reduction is always a primary goal for developing Augmented Reality systems, and the current method provides far from a perfect augmentation. One possibility for achieving this goal is to modify the algorithm to factor in input from multiple tracking systems, similar to a stereo-based camera tracking system. Alternatively, utilizing the camera itself as a means to this end may work in a similar fashion, resulting in a hybrid of the two workflows described in this paper.

Chapter 4

Summary

4.1 Summary and Conclusion

As current methods of minimally invasive surgery leave the surgeon with a distinct spatial disconnect in relation to the surgical site, the work presented in this paper seeks to augment a video view of the site via virtual medical imaging and model data.

The first chapter of this thesis discusses the history of Augmented Reality (AR), as well as applications for the technology in both the past and modern day. Additionally, this chapter explores the various methods of implementation of an AR environment, from the display format to the type of tracking system and methods of registration.

Chapter 2 is a detailed explanation of the workflow necessary to implement a camera-only tracking AR environment. The details included in this chapter discuss the methods of intrinsic and extrinsic calibration, registration of the virtual and real spaces, as well as AR environment evaluation. Intrinsic calibration was performed using Zhang's method [31], while extrinsic calibration was implemented and executed using the ArUco library [19]. The overall accuracy of the system - gauged using the VRE - ranged from 4.57 mm to 17.72 mm, varying with camera position and having a

mean VRE of 9.51 mm. As there were no similar metrics found in literature, the significance of this value was paired with a visual appraisal of the augmentation quality, which was determined to be within acceptable ranges.

Chapter 3 expands the process of reimplementing the method presented in Chapter 2, however using an external tracking system for spatial localization as opposed to using the camera itself. Because of the addition of an external tracking system, specifically the Polaris Spectra optical tracker, the calculation of the extrinsic matrix could no longer be performed using the ArUco library; instead it necessitated the solution handled of the Perspective-n-Point problem. By simultaneously recording the points in the world space and their corresponding points in the image plane, it was possible to solve for the back-projection parameters using the OpenCV library's *solvePnP* function, resulting in the extrinsic matrix that relates the origin of the image coordinate system to the origin of the local coordinate system associated with the rigidly attached tracking sensor mounted onto the camera. Using nearly the same rendering process developed for use in the camera-based implementation, the AR environment was successfully recreated. This method boasted lower error for the VRE - 3.11 mm at the lowest and 13.1 mm at the highest, with a mean RMS error of 8.19 mm.

Despite the successful augmentation of the real camera view with the virtual model generated from a CT scan, this system is far from perfect. Firstly, we identified moderate vertical error in the offset of the external tracking system, particularly associated with certain camera poses, which were constrained by the experimental setup. Second, the display still resides on a standard computer display, a device that lacks the immersive depth of Head-mounted Displays. As the goal of the work was to increase immersion, this

limitation may be viewed as a drawback that can easily be corrected using a head-mounted display such as the Oculus Rift or Vuzix AR/VR eye-wear. Additionally, both tracking systems suffer from obstruction issues - should any object move between the tracked markers and the tracker, the system will cease to function correctly until the obstruction is removed. In a surgical environment where clinicians cannot be expected to be aware of the line-of-sight of the tracker, this could be a major issue. Finally, the current process does not allow for dynamically-updated models (such as a model segmented in real-time from ultrasound data). Any non-rigid, deformable structures relative to the world frame will suffer from inaccuracies unless they are tracked in such a way that any deformations can be handled.

4.2 Contributions

The work presented in this paper demonstrates the process for building an inexpensive, camera-tracked, Augmented Reality guidance environment using an open-source software platform and an affordable recording device. This system was created with the purpose of providing a platform for education, simulation, training, and outreach. Additionally, this work has been extended to demonstrate a similar implementation using a dedicated surgical localization system (the NDI Polaris) and demonstrate its superior performance when compared to the camera-based tracking method. These developments represent the basis for creating more complex video augmented reality applications which rely on the use of 2D video for capturing real-time views of the scene and utilize 3D medical volumes to enhance the visualization of these environments.

4.3 Proposed Future Directions

4.3.1 Improved Video Registration Analysis Metrics

One of the greatest difficulties during this work was determined exactly how to quantify the visual quality of the final augmented scene. Because the final medium is a 2D image, but the input data is in the form of 3D information, the quality of the augmentation is variable dependent upon the position of the camera capturing the scene. This means there is no singular value that can describe the performance of an AR technique. As seen in this paper, results had to be quantified by a range of values, each representing the system's performance at a discrete camera position. The discretized analysis of quality means that no analysis using current methods is truly meaningful unless a slew of other details are provided for each sampled result. To resolve this, it is suggested that the system be constrained in some way during collection of results. In this work, when taking the final augmentation screenshots for analysis the camera was (attempted to be) held at a set distance from the augmentation target. In future works, a structure that positions the camera at known distances from the target may aid in providing context to the results. One suggestion is to create a sort of dolly that moves on a circular track of known diameter, with the world origin at the center of the circle. This would allow for recording multiple images at a fixed distance from the target.

4.3.2 Head-Mounted Display Integration

While the existing system provides functionality to increase visibility in minimally invasive surgeries, it still does not provide a fully immersive environment for the surgeon. Augmentation, unlike laparoscopic visualization, provides a view of the surgical site from outside of the patients body, allowing reference to the real world. However, because information is currently presented on a computer monitor, there is still a spatial disconnect between the surgeons intuitive understanding of where his hands are and what is shown on the screen. To remedy this, we propose integration of a stereoscopic Virtual Reality (VR) headset specifically the Oculus Rift, hereby referred to as the Oculus to augment not just a screen, but the surgeons entire field of vision.

The Oculus Rift is a head mounted display that fits over the eyes. Inside is a small screen positioned behind two lenses, one for each eye. By attaching two cameras to the outside of the Oculus, a live feed of the world can be projected to each eye. These feeds can be augmented using the existing process before being sent to the Oculus, resulting in what, to the user, appears to be an augmentation of their natural vision.

This would require multiple modifications to the existing system. The first of these changes would be to add a second virtual camera to the rendering scene. This is due to the fact that creating a sense of depth that is conducive to a VR environment requires presenting the scene to each of the users eyes at a slightly different angle and position. This second camera would be offset from the first by an amount equivalent to the users interpupillary distance. There are two approaches being considered for this modification. The first is to simply add a second renderer of the same scene with

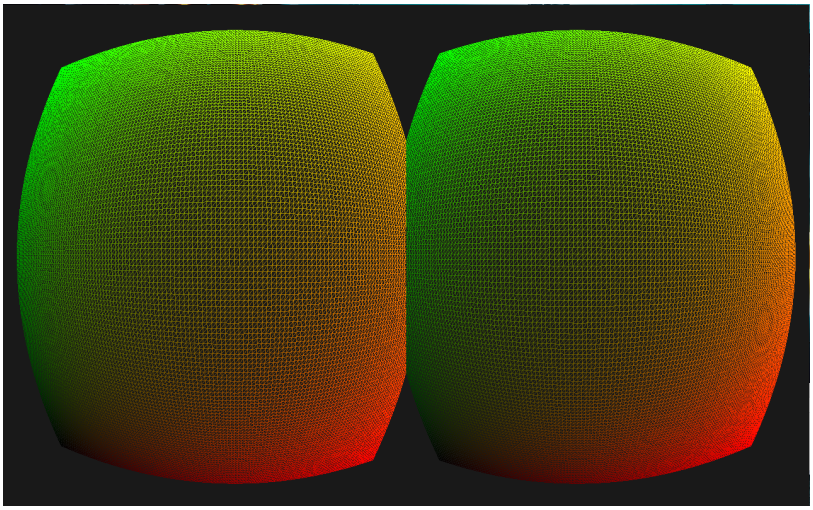

Figure 4.1: Colorized visualization of the Oculus Rift's display mesh used to distort presented images. Courtesy of Alex Benton [34].

a new camera. While this would be more taxing on the hardware running the system, it would be less complicated to implement and would simplify further modifications. The alternative is to instead use a single renderer and update the position of the camera on every frame to alternate between the position of each eye. This technique would be less stressful on hardware but may present issues with frame synchronization between the video feeds to each eye.

Once the views of the two virtual eye positions are obtainable, the existing renderers likely will need to be moved off-screen. The Oculus Rift Development Kit uses mesh-based distortion, and thus provides a special mesh that counteracts the warping caused by the lenses in the headset. The rendered scene is applied as a texture to this mesh. Our current research indicates that the only way to obtain the render as a texture in VTK is to render the scene off-screen (i.e. out of view of the scene camera), capture the scene as an image, apply it to the mesh, and render that mesh in an on-screen renderer. As this is a costly process in terms of performance, it

would be beneficial to research an alternative method that would eliminate the need for off-screen rendering. Once the mesh is rendered, the window needs to be scaled to the native resolution of the Oculus screen and centered in the monitor.

4.3.3 Utilizing Multiple Tracking Systems

In an augmented surgical environment where a surgeon must be unrestricted in their access to the patient, disruption of the augmentation process due to visual obstruction by the surgeon or other element within the space is entirely unacceptable. To remedy this problem, we suggest the addition of functionality within this software to use multiple tracking systems to locate tracked objects. This functionality has been used previously in applications such as SLAM [35]. Additional tracking systems would be placed within the tracked space in such a way that obstructing the line-of-sight to a marker from one tracker does not obstruct all of the other trackers.

Such a configuration would require some form of consolidating the various pose measurements recorded by each tracker into a single measurement. As each measurement contains some inherent error, it will also be necessary to determine how to identify the system error in a meaningful way. Not only does this system provide multiple workspaces in which an object can be detected, thus circumventing obstructions of the line-of-sight in one of the trackers, but it also allows data to be recorded from multiple types of trackers. For example, a magnetic tracker could be paired with an optical tracker in this format. As each tracking system has its own drawbacks and weaknesses, finding a way to utilize various trackers in a single system could greatly improve the robustness of the augmentation process against adverse

elements such as noise sources within different spectra.

4.3.4 Feature-Based Tracking

While the addition of multiple tracking systems can aid in the identification of obscured trackers, it does not resolve the problem experienced by any marker-based AR system: deformation of the tracked object. With our current implementation, the tracked structure is rigid, retaining the same shape that it had when an MR scan was taken of it. However, the human body does not share this quality. Flesh is pliable, blood vessels can move, and organs can shift, expand, or contract. In its current form, the system described in this paper cannot handle these perturbations as the association between the object and its affixed marker is assumed to be rigid and static.

Conceptually, if one were to reduce the object to a cluster of particles, each with a defined position in the scene, it would be possible to affix a marker to track each of these individual elements that comprise the object as if they were an independent entity in the space. Of course, from an implementation standpoint this is problematic. While multiple markers could be affixed to different fiducial points on a deformable object, the size of markers alongside the issue of obstruction doesn't make such an approach feasible. That is why we instead propose a computer vision approach in which these fiducial points (i.e., inherent features) are instead directly tracked by the camera. Feature-based tracking has been implemented previously in various applications such as the OpenCV library with high degrees of success. By applying such a technique to an object such as a lung, which is constantly expanding and contracting, the positions of key features on the organ can be tracked within the tracked space. These points can then be

used in conjunction with a tracked marker attached to the organ to determine the deformation properties of the virtual model. By deforming the model in such a way that the relationship between the feature points and the tracked marker position on the virtual model correspond to that same relationship on the real object, the rest of the augmentation process can proceed unhindered.

Bibliography

[1] WHO. *International Statistical Classification of Diseases and Related Health Problems (International Classification of Diseases)(ICD) 10th Revision - Version:2010*, volume 1. 2010. URL http://apps.who.int/classifications/icd10/browse/2010/en.

[2] Paul Modi, Ansar Hassan, and Walter Randolph Chitwood. Minimally invasive mitral valve surgery: a systematic review and meta-analysis. *European journal of cardio-thoracic surgery : official journal of the European Association for Cardio-thoracic Surgery*, 34(5): 943–952, 2008. ISSN 1873-734X. doi: 10.1016/j.ejcts.2008.07.057.

[3] Jen Jane Liu, Bryan G. Maxwell, Periklis Panousis, and Benjamin I. Chung. Perioperative outcomes for laparoscopic and robotic compared with open prostatectomy using the national surgical quality improvement program (NSQIP) database. *Urology*, 82(3):579–583, 2013. ISSN 00904295. doi: 10.1016/j.urology.2013.03.080. URL http://dx.doi.org/10.1016/j.urology.2013.03.080.

[4] Hop S Tran Cao, Nicole Lopez, David C Chang, Andrew M Lowy, Michael Bouvet, Joel M Baumgartner, Mark a Talamini, and Jason K Sicklick. Improved perioperative outcomes with minimally invasive distal pancreatectomy: results from a population-based analysis. *JAMA surgery*, 149(3):237–43, 2014. ISSN 2168-6262. doi: 10.1001/

jamasurg.2013.3202. URL http://www.ncbi.nlm.nih.gov/pubmed/24402232.

[5] Masahide Kaneko, Fumio Kishino, Kazunori Shimamura, and Hiroshi Harashima. Toward the new era of visual communication. *IEICE Transactions on Communications*, E76-B(6):577–591, 1993. ISSN 09168516. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0027612370{&}partnerID=40{&}md5=cf211359f8d1a133a5021e6b268d27ac.

[6] P.J. Metzger. Adding reality to the virtual. *Proceedings of IEEE Virtual Reality Annual International Symposium*, 1993. doi: 10.1109/VRAIS.1993.380805.

[7] Ziv Yaniv and Cristian A. Linte. Applications of Augmented Reality in the Operating Room. In Woodrow Barfield, editor, *Fundamentals of Wearable Computers and Augmented Reality*, chapter 19, pages 485–518. CRC Press, 2 edition, 2015.

[8] Michol a Cooper, Susan Hutfless, Dorry L Segev, Andrew Ibrahim, Heather Lyu, and Martin a Makary. Hospital level under-utilization of minimally invasive surgery in the United States: retrospective review. *BMJ*, 349(July):g4198, 2014. ISSN 1756-1833. doi: 10.1136/bmj.g4198. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4087169{&}tool=pmcentrez{&}rendertype=abstract.

[9] Madeleine M Keehner, Frank Tendick, Maxwell V Meng, Haroon P Anwar, Mary Hegarty, Marshall L Stoller, and Quan-Yang Duh. Spatial ability, experience, and skill in laparoscopic surgery. *American journal of surgery*, 188(1):71–5, 2004. ISSN 0002-9610. doi: 10.1016/j.amjsurg.2003.12.059. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-3042667198{&}partnerID=tZOtx3y1.

[10] Paul; Haruo Takemura; Akira Utsumi; Fumio Kishino Milgram. Augmented reality: A class of displays on the reality-virtuality continuum. *Telemanipulator and Telepresence Technologies*, 2351: 282–292, 1995. ISSN 0277786X. doi: 10.1117/12.197321. URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=981543.

[11] By Kangdon Lee. Augmented Reality in Education and Training. *TechTrends*, 56(April):13–22, 2012. ISSN 87563894. doi: 10.1007/s11528-012-0559-3.

[12] Ivan E Sutherland. A Head-Mounted, Three-Dimensional Display. In *AFIPS Proceedings of the Fall Joint Computer Conference*, volume Part I, pages 757–764, 1968. doi: 10.1145/1476589.1476686.

[13] Ronald Azuma and Ronald Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997. ISSN 10547460. doi: 10.1.1.30.4999. URL http://scholar.google.com/scholar?q=intitle:A+Survey+of+Augmented+Reality{#}0.

[14] Philip J. Edwards, Andrew P. King, Calvin R. Maurer, Darryl a. De Cunha, David J. Hawkes, Derek L G Hill, Ron P. Gaston, Michael R. Fenlon, a. Jusczyzck, Anthohy J. Strong, Christopher L. Chandler, and Michael J. Gleeson. Design and evaluation of a system for microscope-assisted guided interventions (MAGI). *IEEE Transactions on Medical Imaging*, 19(11):1082–1093, 2000. ISSN 02780062. doi: 10.1109/42.896784.

[15] Simon DiMaio, Mike Hanuschik, and Usha Kreaden. The da Vinci Surgical System. In *Surgical Robotics*, pages 199–217. 2011. ISBN 978-1-4419-1125-4, 978-1-4419-1126-1. doi: 10.1007/978-1-4419-1126-1{_}9.

[16] K Keller, a State, and H Fuchs. Head Mounted Displays for Medical Use. *Journal of Display Technology*, 4(4):468–472, 2008. ISSN 1551-319X. doi: 10.1109/JDT.2008.2001577. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4670082.

[17] Yuichiro Abe, Shigenobu Sato, Koji Kato, Takahiko Hyakumachi, Yasushi Yanagibashi, Manabu Ito, and Kuniyoshi Abumi. A novel 3D guidance system using augmented reality for percutaneous vertebroplasty: technical note. *Journal of neurosurgery. Spine*, 19(4):492–501, 2013. ISSN 1547-5646. doi: 10.3171/2013.7.SPINE12917. URL http://www.ncbi.nlm.nih.gov/pubmed/23952323.

[18] Henry Fuchs and Jeremy Ackerman. Displays for Augmented Reality: Historical Remarks and Future Prospects. 1999.

[19] S Garrido-Jurado, R Muñoz-Salinas, F J Madrid-Cuevas, and M J Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014. ISSN 00313203. doi: 10.1016/j.patcog.2014.01.005.

[20] Terry Peters and Kevin Cleary. *Image-Guided Interventions*, volume 194. 2010. ISBN 9781416029649. doi: 10.2214/AJR.09.3720. URL http://www.springer.com/us/book/9780387738567.

[21] Ziv Yaniv. Which pivot calibration? *Proc. SPIE*, 9415:941527–941529, 2015. ISSN 16057422. doi: 10.1117/12.2081348. URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2081348$\delimiter"026E30F$nhttp://dx.doi.org/10.1117/12.2081348.

[22] Shiqi Li, Chi Xu, and Ming Xie. A robust O(n) solution to the perspective-n-point problem. *TPAMI*, 34(7):1444–1450, 2012. ISSN 01628828. doi: 10.1109/TPAMI.2012.41.

[23] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle. An analytic solution for the perspective 4-point problem. *Proceedings CVPR '89: IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 500–507, 1989. ISSN 1063-6919. doi: 10.1109/CVPR.1989.37893.

[24] Jonathan Fabrizio and Jean Devars. an Analytical Solution To the Perspective-N-Point Problem for Common Planar Camera and for Catadioptric Sensor. *International Journal of Image and Graphics*, 08(01):135–155, 2008. ISSN 0219-4678. doi: 10.1142/

S0219467808003015. URL http://www.worldscientific.com/doi/abs/10.1142/S0219467808003015.

[25] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision*, 81(2):155–166, 2009. ISSN 09205691. doi: 10.1007/s11263-008-0152-6.

[26] Chien Ping Lu, Gregory D Hager, and Eric Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, 2000. ISSN 01628828. doi: 10.1109/34.862199.

[27] Y I Abdel-Aziz and H M Karara. Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry. *Photogrammetric Engineering {&} Remote Sensing*, 81(2):103–107, 2015. ISSN 0099-1112. doi: 10.14358/PERS.81.2.103.

[28] Harvey Rhody. Projective Camera, 2014.

[29] Satoshi Suzuki and Keiichi Abe. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 29(3):396, 1985. ISSN 0734189X. doi: 10.1016/0734-189X(85)90136-7.

[30] David H. Douglas and Thomas K. Peucker. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or

its Caricature. *Classics in Cartography: Reflections on Influential Articles from Cartographica*, 10(2):15–28, 2011. ISSN 0317-7173. doi: 10.1002/9780470669488.ch2.

[31] Zhengyou Zhang Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown\norientations. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1(c):0–7, 1999. doi: 10.1109/ICCV.1999.791289.

[32] Ozgur Guler and Ziv Yaniv. Image-guided navigation: A cost effective practical introduction using the image-guided surgery toolkit (IGSTK). *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pages 6056–6059, 2012. ISSN 1557170X. doi: 10.1109/EMBC.2012.6347375.

[33] Jean-Yves Bouguet. Complete Camera Calibration Toolbox for Matlab, 1999. URL http://www.vision.caltech.edu/bouguetj/.

[34] Alex Benton, Brad Davis, and Karen Bryla. OculusMesh, 2014. URL http://rifty-business.blogspot.com/2014/02/distortion-methods-in-rift-and-their.html.

[35] Michael Kaess and Frank Dellaert. Visual SLAM with a Multi-Camera Rig. *Technology*, 2006. URL http://hdl.handle.net/1853/8726.

[36] Haruo Takemura and Fumio Kishino. Cooperative work environment using virtual workspace. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work - CSCW '92*, number

November, pages 226–232, 1992. ISBN 0897915429. doi: 10.1145/143457.269747. URL http://portal.acm.org/citation.cfm?doid=143457.269747.

[37] Sivakumar Jaikumar, Daniel H. Kim, and Andrew C. Kam. History of minimally invasive spine surgery. *Neurosurgery*, 51(5 SUPPL.):1–14, 2002. ISSN 0148396X. doi: 10.1227/01.NEU.0000030922.20555.BB.

[38] B Preising, T C Hsia, and B. Mittelstadt. A literature review: Robots in medicine, 1991. ISSN 07395175.

[39] L. Soler, S. Nicolau, J. Schmid, C. Koehl, J. Marescaux, X. Pennec, and N. Ayache. Virtual reality and augmented reality in digestive surgery. *ISMAR 2004: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, (Ismar):278–279, 2004. doi: 10.1109/ISMAR.2004.64.

[40] Bernhard Reitinger, Pascal Werlberger, Alexander Bornik, Reinhard Beichel, and Dieter Schmalstieg. Spatial measurements for medical augmented reality. *Proceedings - Fourth IEEE and ACM International Symposium on Symposium on Mixed and Augmented Reality, ISMAR 2005*, 2005:208–209, 2005. doi: 10.1109/ISMAR.2005.53.

[41] Santiago Horgan and Daniel Vanuno. Robots in laparoscopic surgery. *Journal of laparoendoscopic & advanced surgical techniques. Part A*, 11(6):415–9, 2001. ISSN 1092-6429. doi: 10.1089/10926420152761950. URL http://www.ncbi.nlm.nih.gov/pubmed/11814134.

[42] Hui Zhang. Freehand 3D ultrasound calibration using an electromagnetically tracked needle. *Proceedings of SPIE*, 6141: 61412M–61412M–9, 2006. ISSN 0277786X. doi: 10.1117/12.654906. URL http://link.aip.org/link/PSISDG/v6141/i1/p61412M/s1{&}Agg=doi.

[43] Michael R Bax, Rasool Khadem, Jeremy A Johnson, Eric P Wilkinson, and Ramin Shahidi. Calibration and accuracy testing for image-enhanced endoscopy. *TMI*, 85(12):52–56, 2002. ISSN 09269630. doi: 10.3233/978-1-60750-929-5-52.

[44] Andras Lasso, Tamas Heffter, Adam Rankin, Csaba Pinter, Tamas Ungi, and Gabor Fichtinger. PLUS: open-source toolkit for ultrasound-guided intervention systems. *IEEE transactions on biomedical engineering*, 61(10):1–11, 2014. ISSN 1558-2531. doi: 10.1109/TBME.2014.2322864. URL http://www.ncbi.nlm.nih.gov/pubmed/24833412.

[45] Carling L Cheung, Chris Wedlake, J Moore adn S.E. Pautler, Anis Ahmad, and T M Peters. Fusion of stereoscopic video and laparoscopic ultrasound for minimally invasive partial nephrectomy. In *SPIE Medical Imaging 2009: Visualization, Image-Guided Procedures, and Modeling*, volume 7261, pages 7209–7261, 2009. ISBN 9780819475121. doi: 10.1117/12.813741.

[46] Berthold K P Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4): 629, 1987. ISSN 1084-7529. doi: 10.1364/JOSAA.4.000629.

[47] Y Genc, S Riedel, and N Navab. Marker-less Tracking for AR : A Learning-Based Approach. (September), 2002.

[48] M J Daly, H Chan, E Prisman, A Vescan, S Nithiananthan, J Qiu, R Weersink, J C Irish, and J H Siewerdsen. Fusion of intraoperative cone-beam CT and endoscopic video for image-guided procedures. *Proc. SPIE*, 7625:762503–762508, 2010. ISSN 0277786X. doi: 10.1117/12.844212. URL http://dx.doi.org/10.1117/12.844212.

[49] Michael P. DeLisi, Louise A. Mawn, and Robert L. Galloway. Transorbital target localization in the porcine model. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 8671, page 86711S, 2013. ISBN 9780819494450. doi: 10.1117/12.2007943. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84878557440{&}partnerID=tZOtx3y1.

[50] Michael R Bax, Rasool Khadem, Jeremy A Johnson, Eric P Wilkinson, and Ramin Shahidi. Calibration and accuracy testing for image-enhanced endoscopy. *TMI*, 85(12):52–56, 2002. ISSN 09269630. doi: 10.3233/978-1-60750-929-5-52.

[51] Xinyang Liu, He Su, Sukryool Kang, Timothy D Kane, and Raj Shekhar. Application of single-image camera calibration for ultrasound augmented laparoscopic visualization. volume 9415, page 94151T, 2015. ISBN 9781628415056. doi: 10.1117/12.2082194. URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2082194.

[52] Adrian Schneider, Simon Pezold, Andreas Sauer, Jan Ebbing, Stephen Wyler, Rachel Rosenthal, and Philippe C Cattin. Augmented Reality Assisted Laparoscopic Partial Nephrectomy. pages 357–364. 2014. ISBN 978-3-319-10469-0, 978-3-319-10470-6. URL http://link.springer.com.proxy1.lib.uwo.ca/chapter/10.1007/978-3-319-10470-6{_}45.

[53] D Gobbi, R Comeau, and T Peters. Ultrasound Probe Tracking for Real-Time Ultrasound/MRI Overlay and Visualization of Brain Shift. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 920–927, 1999. doi: 10.1007/10704282{_}100. URL http://www.springerlink.com/index/t558k01591342684.pdf.

[54] Marco Feuerstein, Thomas Mussack, Sandro M Heining, and Nassir Navab. Intraoperative laparoscope augmentation for port placement and resection planning in minimally invasive liver resection. *IEEE Transactions on Medical Imaging*, 27(3):355–369, 2008. ISSN 02780062. doi: 10.1109/TMI.2007.907327.

[55] Georg Klein. Visual Tracking for Augmented Reality (Phd Thesis). *System*, (January):182, 2006. URL http://www.robots.ox.ac.uk/{~}gk/publications/Klein2006Thesis.pdf.

[56] Martin Bauer. Tracking Errors in Augmented Reality. page 166 pages, 2007.

[57] Xin Kang, Mahdi Azizian, Emmanuel Wilson, Kyle Wu, Aaron D. Martin, Timothy D. Kane, Craig a. Peters, Kevin Cleary, and Raj Shekhar. Stereoscopic augmented reality for laparoscopic surgery.

Surgical Endoscopy and Other Interventional Techniques, 28(7):2227–2235, 2014. ISSN 14322218. doi: 10.1007/s00464-014-3433-x.